

Fondamenti di Informatica

Ing. Biomedica

Esercitazione n.8

Matrici

Antonio Arena

antonio.arena@ing.unipi.it



UNIVERSITÀ DI PISA

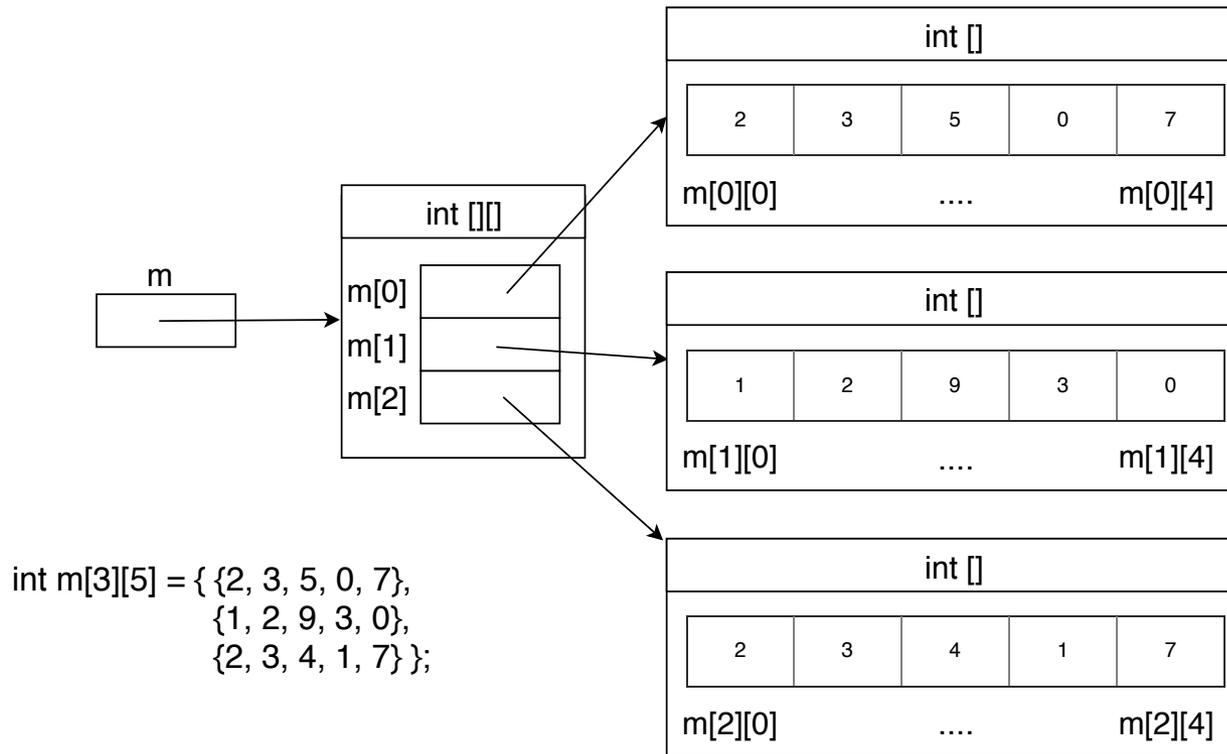


DIPARTIMENTO DI
INGEGNERIA
DELL'INFORMAZIONE

- In C++ viene trattata come array multi-dimensionale. Per esempio, una matrice 4x4 in C++ viene costruita come un array di 4 elementi, e ogni elemento è a sua volta un array di altri 4 elementi.
- *Come si definisce?*
 1. definire subito tutte le dimensioni → `int m[2][4]`
matrice di interi con 2 righe e 4 colonne
 2. non definire **solo** la prima dimensione, *ma va inizializzata tutta subito* →
`int m[][3] = { {3, 5, 7} , {9, 12, 15} };`
→ in questo modo il compilatore capisce che la prima dimensione è 2. Se non si esprime la prima dimensione ma non si inizializza il compilatore genera un errore.

- Per poter passare una matrice come argomento di funzione, la matrice va passata come puntatore ad array.
- La funzione quindi considera la matrice come un un array monodimensionale.
- Alla funzione quindi va passato il puntatore all'elemento (0,0) e le due dimensioni della matrice, ovvero il numero di righe e di colonne.
- **void stampaMatrice (int* m, int r, int c, ...);** → Dichiarazione
- Nel main quindi, supponendo di avere una matrice **int m[10][7];** la chiamata di funzione sarà **stampaMatrice(&m[0][0], 10, 7);**
stampaMatrice((int *)m, 10, 7); → ALTERNATIVA!
- Attenzione, per accedere all'elemento (i,j) della matrice all'interno della funzione, la notazione giusta è: **m[i*c+j]** oppure ***(m+i*c+j)**

Rappresentazione matrice



`int* vm = (int *)m` → LA FUNZIONE VEDE LA MATRICE COSI'

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	3	5	0	7	1	2	9	3	0	2	3	4	1	7

$m[0][0]$ $vm[0*5+2]$ $m[1][0]$ $vm[1*5+2]$ $m[2][0]$ $vm[2*5+4] = m[2][4]$

Esercizio 1

- Scrivere un programma (solo main) che fa le seguenti operazioni:
 - crea un matrice 3x3 di numeri interi
 - legge da tastiera i 9 numeri e li salva nella matrice in ordine di riga, ovvero riempio la prima riga, poi la seconda, e infine la terza
 - Stampa a video la matrice come in esempio
 - Esempio, leggiamo da tastiera i numeri $\langle 1, 10, -3, 7, 0, -2, 4, 6, -7 \rangle$
La matrice stampata sarà la seguente:

1	10	-3
7	0	-2
4	6	-7

Esercizio 1 - soluzione

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int A[3][3];
6
7     for(int i=0;i<3;i++){
8         for(int j=0;j<3;j++){
9             cin >> A[i][j];
10        }
11    }
12
13    for(int i=0;i<3;i++){
14        for(int j=0;j<3;j++){
15            cout << A[i][j] << " ";
16        }
17        cout << endl;
18    }
19    return 0;
20 }
```

Esercizio 1 - variante

- Scrivere due funzioni con prendono come parametro una matrice 3x3.
 - La prima funzione inizializza la matrice leggendo da tastiera i 9 numeri
 - La seconda funzione stampa a video la matrice passata
-
- le intestazioni hanno la seguente forma:
 1. `void leggi_matrice(int m[][3], int r)`
 2. `void stampa_matrice(int m[][3], int r)`

Esercizio 1 – soluzione variante

```
1 #include <iostream>
2 using namespace std;
3
4 const int C = 3;
5 const int R = 3;
6
7 void leggi_matrice(int* m, int r, int c){
8     for(int i=0; i<r ;i++)
9         for(int j=0; j<c; j++)
10            cin >> m[i*c+j];
11 }
12
13 void stampa_matrice(int* m, int r, int c){
14     for(int i=0; i<r ;i++){
15         for(int j=0; j<c; j++){
16             cout << m[i*c +j] << " ";
17         }
18         cout << endl;
19     }
20 }
21
22 int main(){
23     int M[R][C];
24
25     leggi_matrice((int *)M,R,C);
26
27     stampa_matrice(&M[0][0],R,C);
28
29     return 0;
30 }
```

- Scrivere un programma (solo main) che fa le seguenti operazioni:
 - crea un matrice 2x4 di numeri interi
 - legge da tastiera gli 8 numeri e li salva nella matrice in ordine di riga, ovvero riempio la prima riga e poi la seconda
 - Stampa a video la matrice come in esempio
 - Calcola somma e media dei valori nella matrice
 - **Esempio:** leggiamo da tastiera i numeri $\langle 1, 10, -3, 7, 0, -2, 4, 6 \rangle$

L'output sarà:

1	10	-3	7
0	-2	4	6

e poi ... somma = 23, media = 2.875

Esercizio 2 - Soluzione

```
1 #include <iostream>
2 using namespace std;
3
4 const int R = 2;
5 const int C = 4;
6
7 void leggi_matrice(int* m, int r, int c);
8
9 void stampa_matrice(int* m, int r, int c);
10
11 int main(){
12     int M[R][C];
13
14     leggi_matrice((int *)M,R,C);
15
16     stampa_matrice(&M[0][0],R,C);
17
18     double somma = 0, media;
19
20     for(int i=0;i<R;i++)
21         for(int j=0;j<C;j++)
22             somma += M[i][j];
23
24     media = somma / (R*C);
25
26     cout << "Somma: " << somma << endl;
27     cout << "Media: " << media << endl;
28     return 0;
29 }
```

- Scrivere una funzione che prende in ingresso una matrice 2x4 (*stavolta come puntatore a intero!*) e produce la somma e la media dei valori all'interno della matrice.
- intestazione funzione:

```
void sm_matrice(int *m, int r, int c,  
                double& somma, double& media);
```

Esercizio 2 – Soluzione variante

```
1 #include <iostream>
2 using namespace std;
3
4 const int R = 2;
5 const int C = 4;
6
7 void leggi_matrice(int* m, int r, int c);
8
9 void stampa_matrice(int* m, int r, int c);
10
11 void sm_matrice(int* m, int r, int c, double& somma, double& media){
12     somma = 0;
13
14     for(int i=0;i<R;i++)
15         for(int j=0;j<C;j++)
16             somma += M[i*c+j];
17         // potevamo scrivere anche somma += *(m + i*c + j) !!
18     media = somma / (R*C);
19 }
20
21 int main(){
22     int M[R][C];
23
24     leggi_matrice((int *)M,R,C);
25
26     stampa_matrice(&M[0][0],R,C);
27
28     double somma, media;
29     sm_matrice(&M[0][0], R, C, somma, media);
30     cout << "Somma: " << somma << endl;
31     cout << "Media: " << media << endl;
32     return 0;
33 }
```

- Scrivere un programma (solo main) che fa le seguenti operazioni:
 - crea un matrice 2x4 di numeri interi
 - legge da tastiera gli 8 numeri e li salva nella matrice in ordine di riga
 - Stampa a video la matrice
 - Stampa valore massimo nella matrice con i rispettivi indici
 - **Esempio:** leggiamo da tastiera i numeri <1, 10, -3, 7, 0, -2, 4, 6> l'output sarà:

<i>rc</i>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>
<i>0</i>	1	10	-3	7
<i>1</i>	0	-2	4	6

e poi... max = 10, riga = 0, colonna = 1

- Scrivere un programma (solo main) che fa le seguenti operazioni:
 - crea un matrice 3x3 di numeri interi
 - legge da tastiera i 9 numeri e li salva nella matrice in ordine di riga
 - Stampa a video la matrice
 - Calcola e stampa la norma di Frobenius della matrice
- NB: La norma di Frobenius si calcola nel seguente modo:

$$\|A\|_F = \sqrt{\sum_{i,j=0}^n a_{ij}^2}$$

ovvero la radice quadrata della somma dei quadrati di tutti gli elementi di una matrice

Esercizi per casa – 2 esempio

- Esempio, leggiamo da tastiera i numeri
<1, 10, -3, 7, 0, -2, 4, 6, -7>

L'output sarà il seguente:

1	10	-3
7	0	-2
4	6	-7

Norma = 16.2481

- *ovvero* norma =

$$\sqrt{1^2 + 10^2 + (-3)^2 + 7^2 + \dots + 6^2 + (-7)^2} = 16.2481$$



Finally...

Risolvere le richieste dei precedenti 2 esercizi mediante funzioni... come si è visto negli esercizi precedenti