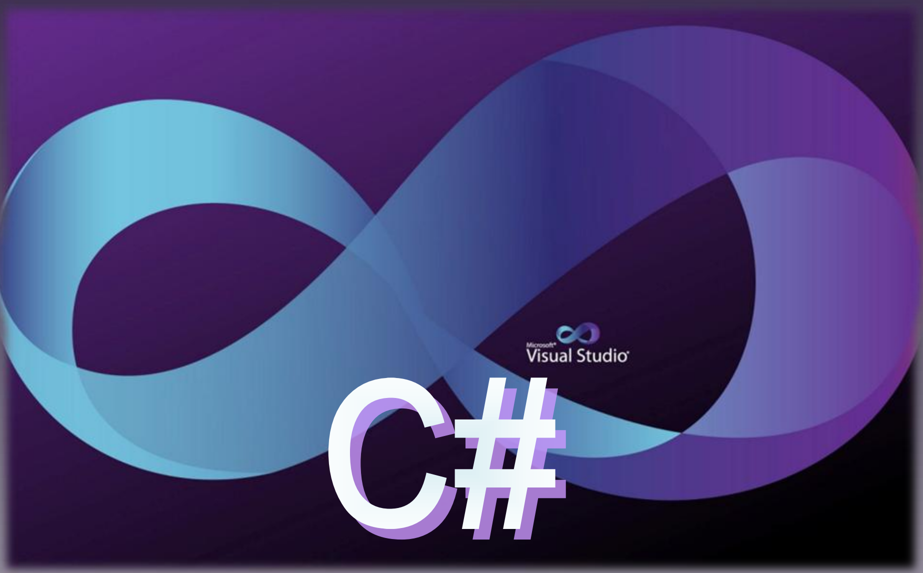
ISTITUTO TECNICO INDUSTRIALE
G. M. ANGIOY
SASSARI



Corso di Programmazione

Le matrici o array multidimensionali

Dispensa 05.04

05-04_Matrici_[ver_15]

Versione del: 07/11/2015
Revisione numero: 15

Prof. Andrea Zoccheddu
Dipartimento di Informatica

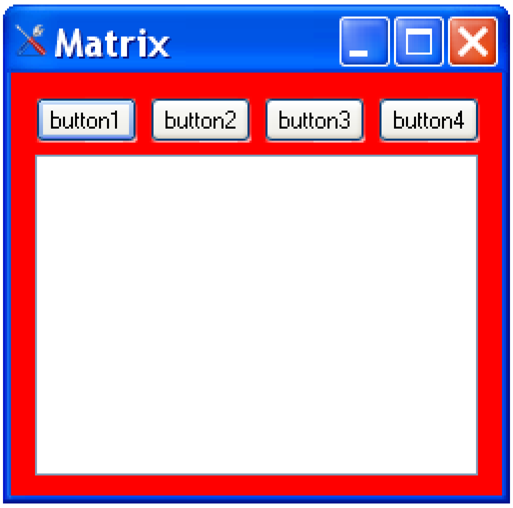
DIPARTIMENTO
INFORMATICA E TELECOMUNICAZIONI

GLI ARRAY A PIÙ DIMENSIONI

LE MATRICI O ARRAY MULTIDIMENSIONALI

INTRODUZIONE ALLE MATRICI

PROGETTO GUIDATO SU MATRICI



- Si prepari un Form1 simile al seguente (c'è un ListBox):
- Preparare le variabili globali seguenti

```
int [,] matrix ;
```

- Doppio clic su **button1** e associare il seguente codice:

```
matrix = new int [3, 5]; //3 righe e 5
colonne
for (int r=0; r<3; r++)
     for (int c=0; c<5; c++)
          matrix [r, c] = -1;
```

- Doppio clic su **button2** e associare il seguente codice:

```
for (int r=0; r<3; r++)
     for (int c=0; c<5; c++)
          matrix [r, c] = (r*10 + c);
```

- Doppio clic su **button3** e associare il seguente codice:

```
for (int r=0; r<3; r++)
     for (int c=0; c<5; c++)
          if ( r == c)
               matrix [r, c] = 1;
          else
               matrix [r, c] = 0;
```

- Doppio clic su **button4** e associare il seguente codice:

```
listBox1.Items.Clear();
for (int r = 0; r < 3; r++)
{
     string s = "";
     for (int c = 0; c < 5; c++)
          s += matrix[r, c] + "\t";
     listBox1.Items.Add(s + "\t");
}
```

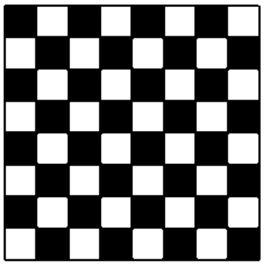
- Prova il progetto.

Nota Bene:

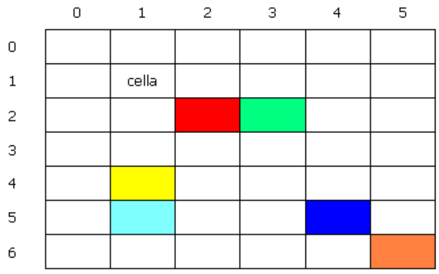
- Nell'esempio è dichiarato un array con due dimensioni: lo si vede dalla virgola tra le parentesi quadre
- L'array a più dimensioni si chiama matrice.
- La matrice va usata con più dimensioni sia per istanziarla che per gli accessi.
- I due indici usati per riferirsi alla cella della matrice si chiamano coordinate di riga e di colonna.

THE MATRIX

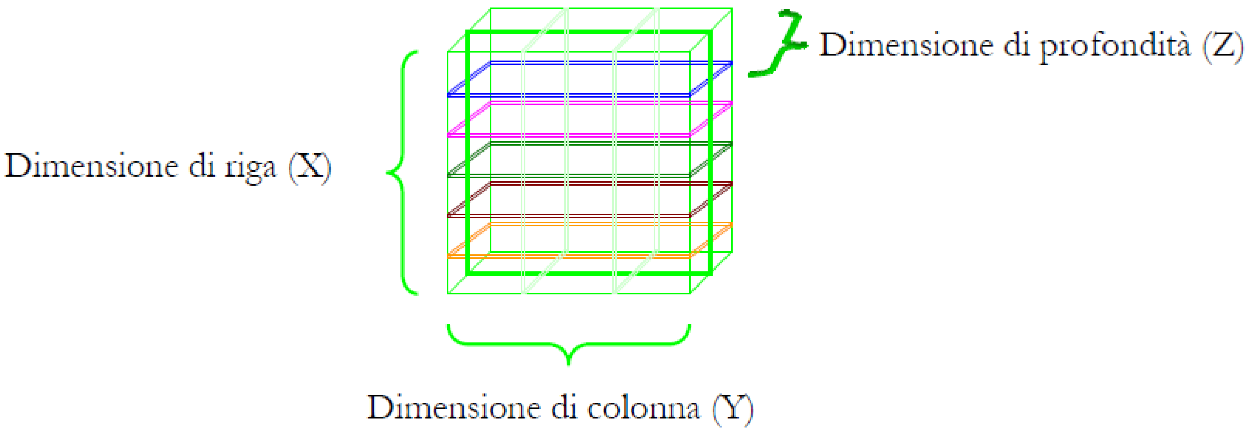
COSA È UNA MATRICE



Le matrici sono un tipo di dato analogo strutturato. Le matrici sono array, ma le celle sono disposte su più dimensioni. Una matrice con due dimensioni, per esempio, è la scacchiera (figura a lato); la scacchiera ha 8 righe e 8 colonne per cui ha un totale di 64 celle. Le matrici dei linguaggi di programmazione possono avere un differente numero di righe e/o di colonne. Come per gli array occorre numerare le righe e le colonne, in modo da individuare ogni cella. Per esempio la figura a sinistra mostra una matrice con 7 righe (numerate da zero a sei) e sei colonne (numerate da zero a cinque). Poiché la matrice ha due dimensioni sono necessarie due coordinate per individuare la cella, come negli schemi della battaglia navale o negli scacchi. Per esempio la cella di colore rosso ha coordinate 2;2; Invece la cella blu ha coordinate 5;4. La prima cella della matrice ha sempre coordinate 0;0. L'ultima cella della figura ha coordinate 6;5.



In alcuni casi posso definire delle matrici con più di due dimensioni, per esempio tridimensionali o multidimensionali. Nella figura seguente è disegnata una matrice con tre dimensioni, con tre indici diversi.



Potrei riferirmi alla cella di coordinate [2 , 1 , 3] per indicare la cella della seconda riga, prima colonna, terza fila. Se occorre è possibile dichiarare matrici con 4 o più dimensioni. Se una matrice ha N dimensioni ogni cella deve essere individuata esattamente con N indici.

Una matrice a due dimensioni potrebbe essere come la seguente figura. Per verificare che hai compreso il meccanismo degli indici, prova a considerare le seguenti espressioni:

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

M [0 , 0] = 2;

M [1 , 0] = 4;

M [0 , 1] = 3;

M [2 , 2] = 10;

M [3 , 0] = 11;

M [0 , 3] = ? ; la cella 0, 3 non esiste in questa matrice!!! È importante che tu abbia compreso il significato e l'uso delle coordinate di identificazione di una cella.

A COSA SERVE UNA MATRICE

Una matrice può servire per rappresentare dati di un gioco (come la dama), la tabella di un orario (come quello scolastico) o qualsiasi struttura che utilizzi più coordinate per riferirsi ad un elemento del problema.

Le matrici si usano in diversi ambiti: matematica, grafica, giochi ecc. Per esempio, la dichiarazione di una matrice per giocare a campo minato potrebbe essere la seguente:

```
int[ , ] bombe = new int[12 , 20];
int[ , ] campo = new int[12 , 20];
```

Che descrive due matrici, la prima contenente bombe e celle vuote; la seconda bandiere, numeri e celle scoperte e celle da scoprire. In questo caso possiamo usare una codifica; per esempio:

- nella matrice `bombe` il valore 0 indica una cella senza bomba, mentre il valore 1 indica nella cella la presenza di una bomba.
- nella matrice `campo` il valore 0 indica una cella da scoprire, senza bandiere; il valore 1 indica una cella scoperta; il valore 2 si riferisce a una cella con la bandiera della bomba e infine il valore 3 indica un punto di domanda sopra la cella.

Per realizzare un gioco della dama potremmo dichiarare una matrice come la seguente:

```
int[ , ] dama = new int[8 , 8];
```

dove (usando solo interi compresi tra -2 e +2):

-1 → pedina nera *0 → casella vuota* *1 → pedina bianca*

-2 → dama nera *2 → dama bianca*

LE MATRICI IN VISUAL C#

DICHIARAZIONE

Una matrice viene dichiarata in modo del tutto analogo agli array; si separano però con delle virgole le dimensioni (gli indici) necessari. Per esempio:

```
int[ , ] matrix = new int[5 , 10];
string[ , ] orario = new int[7 , 6];
```

spesso è preferibile usare delle costanti per dichiarare la dimensione della matrice; per esempio:

```
const int NR = 10;
const int NC = 5;
int[,] M = new int[NR, NC];
```

RICAVARE LE DIMENSIONI «AL VOLO»

Non sempre è possibile sapere in anticipo le dimensioni della matrice; per ottenere le dimensioni dei una matrice è possibile usare i metodi GetLenght(...), specificando tra parentesi tonde la dimensione desiderata. Per esempio:

```
int righe = matrix.GetLength(0);
int colon = matrix.GetLength(1);
MessageBox.Show("" + righe + "; " + colon);
```

che mostra in un messaggio le dimensioni della matrice matrix.

ACCESSO ALLE CELLE IN SCRITTURA

Anche l'uso in scrittura è analogo al vettore ma si devono usare due indici. Per esempio:

```
matrix[0,0] = 12; //assegna 12 alla prima cella della matrice matrix
orario[0,0] = "ciao"; //assegna "ciao" alla prima cella della matrice orario
matrix[1,3] = 17; //assegna 17 alla cella della riga 1 e colonna 3 di matrix
orario[6,5] = "Storia"; //assegna "Storia" alla ultima cella di orario
```

ACCESSO ALLE CELLE IN LETTURA

Anche l'uso in lettura è analogo al vettore ma si devono usare due indici. Per esempio:

```
int a = matrix[0,0]; //assegna il valore della prima cella di matrix ad a
string s = orario[0,0]; //assegna il valore della prima cella di orario ad s
```

È possibile anche usarle sia in lettura che in scrittura in una istruzione / operazione. Per esempio:

```
matrix[1,3]++; //incrementa la cella della riga 1 e colonna 3
orario[6,5] += " moderna"; //concatena " moderna" nella ultima cella di
orario
```

ALGORITMI ELEMENTARI

Come per i vettori è possibile creare degli algoritmi per usare le matrici. Vediamo qualcuno. Quando ritieni di aver capito, crea un programma e prova ad associare ad un pulsante qualcuno dei seguenti algoritmi:

INIZIALIZZAZIONE DI UNA MATRICE BIDIMENSIONALE

Un algoritmo per porre valori casuali in una matrice di interi:

```
Random Rnd = new Random();
for (int r = 0; r < NR; r++)
   for (int c = 0; c < NC; c++)
      M[r, c] = Rnd.Next(100,1000);
```

Commento: uso due indici che chiamo r (come righe) e c (come colonne). La matrice è M, quella dichiarata in precedenza. La matrice M ha NR righe ed NC colonne (sono le costanti dichiarate prima). Controlla attentamente come funzionano i cicli. Un altro algoritmo per porre valori in una matrice di interi è:

```
for (int r = 0; r < NR; r++)
   for (int c = 0; c < NC; c++)
      M[r, c] = ((r * 10) + c );
```

le celle prendono valori che dipendono dal loro indice.

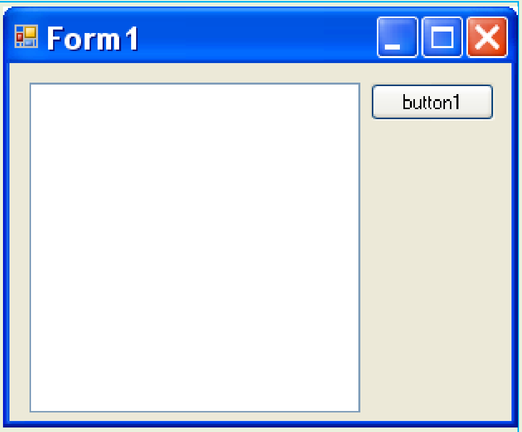
Esercizio

Dichiara una matrice adatta alla tavola pitagorica con 10x10 celle. Crea un algoritmo che crea in automatico la tabellina pitagorica.

VISUALIZZAZIONE

Prepara un Form1 come nella figura e associa al pulsante un algoritmo per porre valori casuali in una matrice di interi e visualizzarla in un listBox1:

```
Random Rnd = new Random();
for (int r = 0; r < NR; r++)
   for (int c = 0; c < NC; c++)
      M[r, c] = Rnd.Next(100,1000);
listBox1.Items.Clear();
for (int r = 0; r < NR; r++)
{
   string s = ""; //stringa vuota
   for (int c = 0; c < NC; c++)
   {
      if (M[r,c] < 10)
         s += " 0" + M[r, c];
      else
         s += " " + M[r, c];
   }
listBox1.Items.Add(s);
}
```



Prova tu ad aggiungere pulsanti e a far loro eseguire i seguenti compiti:

- **Somma** di tutti gli elementi di una matrice (visualizza il risultato in una etichetta)
- **Media** di tutti gli elementi di una matrice (visualizza il risultato in una etichetta)
- **Minimo** tra tutti gli elementi di una matrice (visualizza il risultato in una etichetta)
- **Massimo** tra tutti gli elementi di una matrice (visualizza il risultato in una etichetta)
- **Posizione** (coordinate) della cella che contiene il valore minimo tra tutti gli elementi di una matrice (visualizza il risultato in una etichetta)

MATRICI QUADRATE

Alcune matrici hanno indici differenti per righe e per colonne e sono rappresentate con una forma rettangolare. Alcune matrici però hanno la stessa dimensione di riga e di colonna; la loro rappresentazione è un quadrato, da cui il nome matrice quadrata. Un esempio di matrice quadrata è la scacchiera della dama.

Le matrici quadrate hanno permesso di sviluppare algoritmi particolari e poi una algebra molto interessante. In una matrice quadrata è interessante osservare che essa ammette due diagonali ed esistono degli algoritmi riferiti alle diagonali della matrice.

Un esempio è proposto nelle seguenti figure: La diagonale della matrice evidenziata nella figura, formata dalle celle che hanno stesso indice di riga e di colonna, è detta diagonale principale. Per la diagonale è possibile realizzare gli stessi algoritmi già visti per i vettori ad una dimensione. Verifica che siano vere le seguenti espressioni:

13	5	5	4	1
2	**7**	14	6	7
11	5	**8**	3	0
10	19	7	**2**	9
2	21	13	17	**1**

M [1 , 1] = 13;
M [2 , 2] = 7;
M [3 , 3] = 8;
M [4 , 4] = 2;
M [5 , 5] = 1 ;

ESERCIZI DI RIPASSO SULLE MATRICI QUADRATE

Propongo qui un breve riepilogo degli algoritmi da realizzare sulla diagonale principale:

- Somma degli elementi della diagonale principale
- Media della diagonale principale
- Prodotto degli elementi della diagonale principale
- Minimo, massimo, posizione del minimo e del massimo della diagonale principale
- Ordinamento della diagonale principale
- Ci sono zeri sulla diagonale?
- Ricerca di un elemento sulla diagonale
- Nella diagonale ci sono solo zeri?

Come esempio svolgo il primo esercizio; gli altri sono lasciati per esercizio. Osserva che NDIM è la dimensione della matrice ovvero il numero delle righe (ed ovviamente coincide col numero delle colonne):

```
int somma = 0;
for (int k = 0; k < NDIM; k++)
    somma += M[k, k];
```

come si vede l'indice è lo stesso per la riga e per la colonna della cella, poiché è sulla diagonale.

Oltre alla diagonale principale esiste anche la diagonale secondaria, formata dalle celle evidenziata nella figura seguente: Propongo qui un breve riepilogo degli algoritmi da realizzare sulla diagonale secondaria (appena più difficili per individuare gli indici di riga e colonna):

13	5	5	4	**1**
2	7	14	**6**	7
11	5	**8**	3	0
10	**19**	7	2	9
2	21	13	17	1

- Somma degli elementi della diagonale secondaria
- Media della diagonale secondaria
- Prodotto degli elementi della diagonale secondaria
- Minimo, massimo, posizione del minimo e del massimo della diagonale secondaria
- Ordinamento della diagonale secondaria

ALGEBRA DELLE MATRICI

Esiste una algebra delle matrici sviluppata e ricca di algoritmi, alcuni dei quali assai complicati. Per il momento ci limiteremo solo ai più importanti algoritmi sulle matrici:

SOMMA TRA MATRICI:

1	5	0	4	1
2	7	1	6	7
1	5	8	3	0
0	0	3	2	9
2	10	0	1	1

+

2	2	1	3	0
8	3	1	4	0
3	2	3	1	0
1	4	2	2	2
1	8	0	0	0

=

3	7	1	7	1
10	10	2	10	14
4	7	11	6	0
1	4	5	4	11
3	18	0	1	1

La somma di matrici è semplice ed è analoga alla somma di vettori: ciascuna cella della matrice somma assume il valore della somma delle celle corrispondenti nelle due matrici addendo. L'algoritmo richiede due for annidati ed è lasciato per esercizio. Come si può intuire, la somma di matrici non si riferisce alle sole matrici quadrate ma anche a matrici non quadrate, purché gli addendi abbiano la stessa dimensione. La matrice risultato ha la stessa dimensione delle matrici addendo. Di solito si usa la notazione (RxN) per indicare una matrice con R righe e N colonne. La somma di sopra pertanto potrebbe essere rappresentato tra classi di matrici 5x5.

MATRICE NULLA

La matrice Nulla O è una matrice composta di soli zeri.

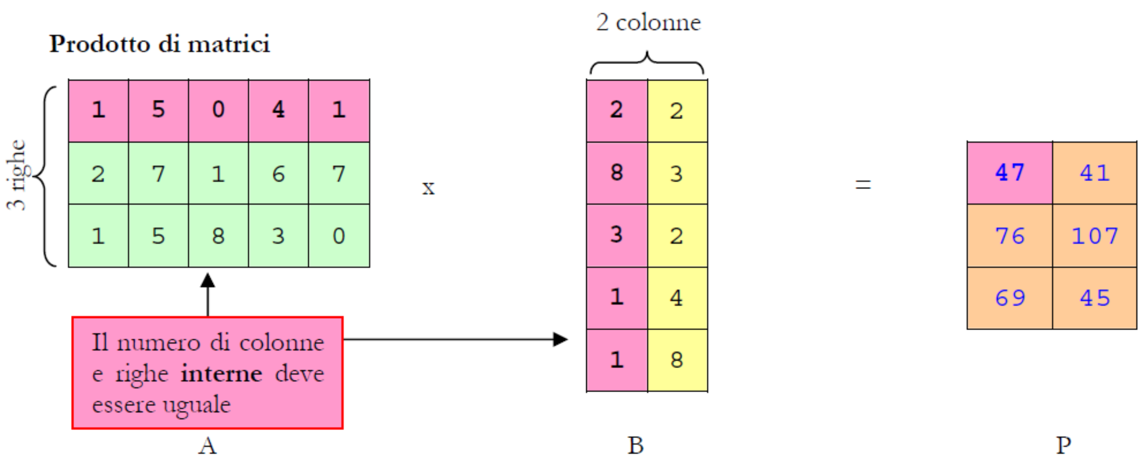
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

La matrice Nulla è l'elemento neutro rispetto alla somma, ovvero per qualsiasi matrice M si ha che: **M + O = O + M = M**.

PRODOTTO TRA MATRICI

Il prodotto tra matrici si può fare solo se la prima matrice fattore ha un numero di colonne esattamente uguale al numero di righe della seconda matrice fattore. La matrice risultante ha un numero di righe identico al numero di righe della matrice primo fattore; inoltre la matrice risultante ha un numero di colonne identico al numero di colonne della matrice secondo fattore.

Nella figura qui mostrata, la prima matrice ha dimensione (3x5) e la seconda ha dimensione (5x2): tra loro si possono moltiplicare e la matrice risultante ha dimensione (3x2). Come si ottengono i valori delle celle della matrice prodotto? Consideriamo la cella della matrice prodotto di coordinate h e k. La cella P[h,k] si ottiene col prodotto scalare del vettore riga h della prima matrice e del vettore colonna k della seconda matrice.



Ad esempio la cella P[1,1] della matrice prodotto si ottiene moltiplicando la prima riga di A e la prima colonna di B. In pratica si tratta del seguente valore:

P[1,1] = 1x2 + 5x8 + 0x3 + 4x1 + 1x1 = 2 + 40 + 0 + 4 + 1 = 47.

MATRICE IDENTITÀ

La matrice Identità è una matrice quadrata composta di soli zeri tranne sulla diagonale principale dove ci sono soli uni. La matrice Identità è l'elemento neutro rispetto al prodotto, ovvero per qualsiasi matrice M si ha che: **M x I = I x M = M.**

1	0	0	0
0	**1**	0	0
0	0	**1**	0
0	0	0	**1**

MATRICE TRASPOSTA

Data una matrice M, la sua matrice Trasporta M^T è una matrice ottenuta da M scambiando i suoi elementi simmetricamente rispetto alla sua diagonale principale. Come si vede gli elementi delle righe della matrice diventano gli elementi delle colonne della sua trasposta.

1	2	0	3
4	**5**	**6**	**7**
8	9	**0**	1
2	3	4	**9**

1	**4**	8	2
2	**5**	9	3
0	**6**	**0**	4
3	**7**	1	**9**

DETERMINANTE DI MATRICE

In algebra lineare, il determinante è una funzione che associa ad ogni matrice quadrata A uno scalare che ne sintetizza alcune proprietà algebriche. In particolare se la matrice ha ordine 1, il determinante di A è semplicemente il valore del suo unico elemento:

$$\det(A) := a_{1,1}$$

Se la matrice ha ordine 2, il determinante si ottiene con la seguente formula:

$$\det(A) := a_{1,1}a_{2,2} - a_{2,1}a_{1,2}$$

Se la matrice ha ordine 3, il determinante si ottiene con la seguente formula:

$$\det(A) := a_{1,1}a_{2,2}a_{3,3}+a_{1,3}a_{2,1}a_{3,2}+a_{1,2}a_{2,3}a_{3,1}-a_{1,3}a_{2,2}a_{3,1}-a_{1,1}a_{2,3}a_{3,2}-a_{1,2}a_{2,1}a_{3,3}$$

Nelle formule, la lettera a minuscola indica l'elemento della matrice e i numeri posti a pedice sono le coordinate di riga e colonna (dove la prima riga e la prima colonna sono numerate con 1).

ATTENZIONE: in Visual Studio le coordinate delle matrici iniziano con l'indice zero, ma nella teoria generale delle matrici invece le prime coordinate si indicano con 1. Quando si programma in VC# occorre fare attenzione a usare i numeri correttamente. Per esempio il determinante di una matrice di ordine zero sarà: M[0,0] (uso zero invece di uno).

Ad esempio:

M =

9

Allora il determinante di M è:

det (M) = **9**

P =

1	**4**
2	**5**

Allora il determinante di P è:

det (P) = 1x5 – 4x2 = 5 – 8 = **– 3**

Q =

1	**4**	**0**
2	**1**	**5**
0	**3**	**1**

Allora il determinante di Q è:

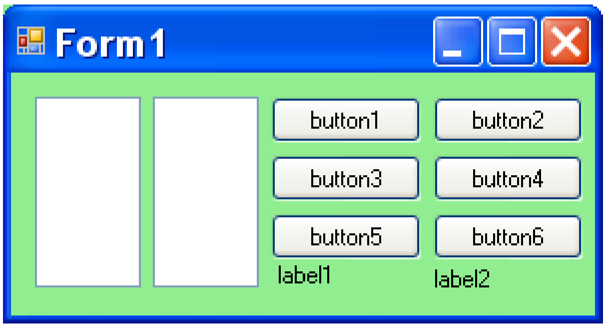
det (Q) = 1x1x1 + 0x2x3 + 4x5x0 – 0x1x0 – 1x5x3 – 4x2x1 =

= 1 + 0 + 0 – 0 – 15 – 8 = **– 22**

ESERCIZI

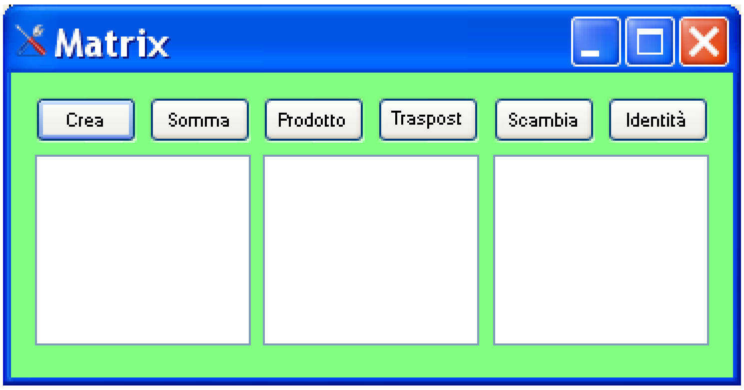
ESERCIZI SU: MATRICI

ESERCIZIO 1. ALGORITMI SULLE MATRICI



- Dichiarare due matrici di interi.
- **Pulsante1** : crea le due matrici quadrate di ordine 10 e inserisce nella prima valori casuali e prepara la seconda come identità.
- **Pulsante2** : visualizza le due matrici nelle ListBox
- **Pulsante3** : mette nella seconda matrice la trasposta della prima
- **Pulsante4** : modifica la prima matrice in modo da trasporla (non usare una matrice di appoggio)
- **Pulsante5** : inverte i segni della seconda matrice
- **Pulsante6 :** crea una nuova matrice ottenuta dal prodotto delle prime due e la visualizza nella prima ListBox; pulisce la seconda ListBox.

ESERCIZIO 2. ANCORA ALGORITMI SULLE MATRICI



- Dichiarare tre matrici di interi di ordine 10. Siano i loro nomi M, N e P. I pulsanti fanno:
- **Crea** : inserisce in M ed in N valori casuali diversi e le mostra nelle prime due ListBox
- **Somma** : pone in P la somma di M+N e la visualizza nella terza ListBox
- **Prodotto**: pone in P il prodotto di MxN e la visualizza nella terza ListBox
- **Trasposta**: traspone la seconda matrice e la visualizza nella seconda ListBox
- **Scambia**: scambia le prime due matrici e le visualizza nelle prime due ListBox
- **Identità**: modifica M per farla diventare la matrice Identità e infine la visualizza nella prima ListBox

ESERCIZIO 3. DI NUOVO ALGORITMI SULLE MATRICI

- Dichiarare tre matrici di interi di ordine 10. Siano i loro nomi M, N e P.
- **Crea** : inserisce in M ed in N valori casuali diversi e le mostra nelle prime due ListBox.
- **Red** : ruota gli elementi di M in senso antiorario, poi la mostra in listBox1
- **Cyan** : ruota gli elementi di M in senso orario, poi la mostra in listBox1
- **Trasposta** : verifica se M ed N hanno un valore in comune e mostra un messaggio
- **Droid** : pone in ogni cella di P il massimo delle rispettive celle di M e N
- **Kalk** : mostra in un messaggio il valore medio degli elementi della diagonale principale di P

ESERCIZIO 4. INFINE ALGORITMI SULLE MATRICI

- Dichiarare tre matrici di interi di ordine 8. Siano i loro nomi M, N e P.
- **Crea** : inserisce in M ed in N valori casuali diversi e le mostra nelle prime due ListBox; infine crea P come prodotto di MxN e la mostra nella terza ListBox3
- **Cerchio Rosso**: somma gli elementi della prima colonna di M e li visualizza in un messaggio
- **Cerchio Azzurro**: somma gli elementi della prima riga di M e li visualizza in un messaggio
- **Bolle Colorate**: somma gli elementi della diagonale secondaria di M
- **Droid** : pone in ogni cella di P il minimo delle rispettive celle di M e N
- **Kalk** : mostra in un messaggio la somma degli elementi che superano il valore medio degli elementi della diagonale principale di P

SOMMARIO