

Esempio Pilota: InputOutputDiUnaMatrice

Definizione della Form

Codice della Form

```
using System;
using System.Windows.Forms;
using Microsoft.VisualBasic;
```

```
namespace InputOutputDiUnaMatrice
{
    public partial class frmAvvio : Form
    {
        public frmAvvio()
        {
            InitializeComponent();
        }
    }
}
```

```
// Nota Bene: la ListBox ha tipo di carattere "Courier New" i cui
// caratteri occupano tutti una stessa larghezza e garantiscono
// un corretto incolonnamento dei dati visualizzati.
```

```
// Dichiarazione della Matrice (max 10x10):
```

```
// Costanti per indicare la dimensione MASSIMA della matrice ...
const int MaxNR = 10;
const int MaxNC = 10;
```

```
// Dichiarazione della matrice ...
int [, ] Mat = new int [ MaxNR , MaxNC ];

// Variabili che indicano il numero di righe e colonne
// realmente utilizzate per la memorizzazione dei dati ...
int NR = 0;
int NC = 0;
```

```
private void plsAvvia_Click(object sender, EventArgs e)
{
```

```
    // Acquisisci in Numero di Righe e di Colonne richiesto ...
    NR = Convert.ToInt16(txtNR.Text);
    NC = Convert.ToInt16(txtNC.Text);
```

```
    // Assicurati che le dimensioni richieste non superino
    // le dimensioni massime previste per la matrice ...
```

```
    if ((NR <= MaxNR) && (NC <= MaxNC))
    {
```

```
        // Ciclo esterno R che scandisce le RIGHE della matrice ...
```

```
        for ( int R = 0; R <= NR-1; R++ )
```

```
        {
```

```
            // Ciclo interno C che, per ogni riga R, scandisce le COLONNE
```

```
            for ( int C = 0; C <= NC-1; C++ )
```

```
            {
```

```
                // Input dell'elemento Mat(R,C) ... riga R, colonna C ...
```

```
                Mat [R , C] = Convert.ToInt16(
                    Interaction.InputBox("Elemento " + R + ", " + C + ".:"));
```

```
            }
        }
    }
}
```

```
else
```

```
    MessageBox.Show(
        "Dimensioni massime: " + MaxNR + ", " + MaxNC);
}
```

```
private void plsVisualizza_Click(object sender, EventArgs e)
{
```

```
    // Ciclo esterno R che scandisce le RIGHE della matrice ...
```

```
    for ( int R = 0; R <= NR-1; R++ )
```

```
    {
```

```
        // Variabile per comporre la riga di dati da aggiungere alla ListBox ...
```

```
        string Riga = "";
```

```
        // Ciclo interno C che, per ogni riga R, scandisce tutte le COLONNE ...
```

```
        for ( int C = 0; C <= NC-1; C++ )
```

```
        {
```

```
            // Aggiungi alla Riga l'elemento Mat(R,C) ... riga R, colonna C ...
```

```
            // (per garantire l'allineamento, il metodo PadLeft aggiunge spazi
```

```
            // a sinistra in modo che il dato occupi esattamente 4 caratteri )
```

```
            Riga += Mat[R, C].ToString().PadLeft(4);
```

```
        }
    }
}
```

```
    // Aggiungi, alla ListBox, la Riga appena costruita ...
```

```
    lstDatiMat.Items.Add(Riga);
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```