

Esempio Pilota: **CarattInsElimVoti**

Si consenta l'input delle **Materie** e dei relativi **Voti** di un alunno memorizzandoli in due vettori **VetMaterie** e **VetVoti** (max 10), consentendo di **aggiungere, inserire ed eliminare** le materie dall'elenco, di calcolarne il **numero di insufficienze** e di verificare se l'alunno è promosso, ossia se è vero o no che **tutti i voti sono sufficienti**.

Definizione della Form

Codice della Form

```
using System;
using System.Windows.Forms;
```

```
namespace CarattInsElimVoti
```

```
{
    public partial class Form1 : Form
```

```
{
    public Form1()
    {
        InitializeComponent();
    }
}
```

```
// dichiarazioni globali (i due vettori) ...
```

```
const int Max = 10;
```

```
string [] VetMaterie = new string [ Max ];
int [] VetVoti = new int [ Max ];
```

```
int N = 0;
```

```
// sottoprogrammi-evento (click sui vari pulsanti) ...
```

```
private void plsAggiungiAllaFine_Click
    (object sender, EventArgs e)
{
    Aggiungi ( txtMateria.Text,
                Convert.ToInt32 ( txtVoto.Text ) );

    Visualizza ( lblElenco );
}
```

```
private void plsInserisci_Click (object sender, EventArgs e)
{
    Inserisci ( txtMateria.Text,
                Convert.ToInt32 ( txtVoto.Text ),
                Convert.ToInt32 ( txtPosIns.Text ) );

    Visualizza ( lblElenco );
}
```

```
private void plsElimina_Click (object sender, EventArgs e)
{
    Elimina ( Convert.ToInt32 ( txtPosElim.Text ) );
    Visualizza ( lblElenco );
}
```

```
private void plsNumeroInsufficienze_Click
    (object sender, EventArgs e)
{
    lblNumInsuff.Text = NumeroDiInsufficienze( ).ToString( );
}
```

```
private void plsVerificaPromozione_Click
    (object sender, EventArgs e)
{
    if ( VotiTuttiSufficienti( ) )
        lblVerificaPromozione.Text = "SI";
    else
        lblVerificaPromozione.Text = "NO";
}
```

```
// sottoprogrammi che operano sui vettori ...
```

```
private void Aggiungi (string Materia, int Voto)
{
    if ( N < Max )
    {
        VetMaterie [ N ] = Materia;
        VetVoti [ N ] = Voto;

        N++;
    }
    else
        MessageBox.Show ("Impossibile inserire piu' di "
                        + Max + " materie.");
}
```

```

private void Inserisci (string Materia, int Voto, int PosIns)
{
    if (N < Max)
    {
        if ( (PosIns >= 0) && (PosIns <= N - 1) )
        {
            // sposta in avanti gli elementi da PosIns a N-1 ...
            for (int K = N - 1; K >= PosIns; K--)
            {
                VetMaterie [K + 1] = VetMaterie [K];
                VetVoti [K + 1] = VetVoti [K];
            }

            // inserisci i nuovi dati in posizione PosIns ...
            VetMaterie [PosIns] = Materia;
            VetVoti [PosIns] = Voto;

            // aggiorna N ...
            N++;
        }
        else
            MessageBox.Show( "Impossibile inserire in posizione "
                + PosIns);
    }
    else
        MessageBox.Show( "Impossibile inserire piu' di "
            + Max + " materie.");
}

```

```

private void Elimina (int PosElim)
{
    if ( (PosElim >= 0) && (PosElim <= N - 1) )
    {
        // sposta all'indietro gli elementi da PosElim+1 a N ...
        for (int K = PosElim + 1; K <= N - 1; K++)
        {
            VetMaterie [K - 1] = VetMaterie [K];
            VetVoti [K - 1] = VetVoti [K];
        }

        // aggiorna N ...
        N--;
    }
    else
        MessageBox.Show( "Impossibile eliminare in posizione "
            + PosElim);
}

```

```

private void Visualizza (Label lbl)
{
    lbl.Text = "";

    for (int K = 0; K <= N - 1; K++)
    {
        lbl.Text = lbl.Text + VetMaterie [K] + ". " +
            VetVoti [K] + "\n";
    }
}

```

```

private int NumeroDiInsufficienze ( )
{
    // algoritmo che opera solo su elementi del vettore
    // AVENTI UNA DATA CARATTERISTICA

    int ContNumerolInsufficienze = 0;

    for (int K = 0; K <= N - 1; K++)
    {
        if ( VetVoti [K] < 6 ) // caratteristica: "voto insufficiente"
        {
            ContNumerolInsufficienze++;
        }
    }

    return ContNumerolInsufficienze;
}

```

```

private bool VotiTuttiSufficienti ( )
{
    // algoritmo che VERIFICA SE TUTTI GLI ELEMENTI
    // DI UN VETTORE HANNO UNA DATA CARATTERISTICA

    bool TuttiSufficienti = true;

    for (int K = 0; K <= N - 1; K++)
    {
        if ( VetVoti [K] < 6 ) // caratteristica: "voto insufficiente"
        {
            TuttiSufficienti = false;
            break;
        }
    }

    return TuttiSufficienti;
}

```

```

}
}

```