Seating Chart

Enter a Row   4

Enter a Column   3

Seat Price $475.00

Columns

	0	1	2	3
Rows 0	☐	☐	☐	☐
1	☐	☐	☐	☐
2	☐	☐	☐	☐
3	☐	☐	☐	☐
4	☐	☐	☐	☐
5	☐	☐	☐	☐

Display Price   Exit

```csharp
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Seating_Chart
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void displayPriceButton_Click(object sender, EventArgs e)
        {
            // Variables for the selected row and column
            int row, col;

            const int MAX_ROW = 5;
            const int MAX_COL = 3;

            // Create an array with the seat prices
            decimal[,] prices =
            {
                { 450m, 450m, 450m, 450m },
                { 425m, 425m, 425m, 425m },
                { 400m, 400m, 400m, 400m },
                { 375m, 375m, 375m, 475m },
                { 375m, 375m, 375m, 475m },
                { 350m, 350m, 350m, 450m }

            };
            // Get the selected row number.
            if (int.TryParse(rowTextBox.Text, out row))
            {
                // Get the selected column number.
                if (int.TryParse(colTextBox.Text, out col))
                {
                    // Make sure the row is within range.
                    if (row >= 0 && row <= MAX_ROW)
                    {
                        if (col >= 0 && col <= MAX_COL)
                        {
                            // Display the selected seat's price.
                            priceLabel.Text = prices[row, col].ToString("c");
                        }
                        else
                        {
                            // Error message for invalid column.
                            MessageBox.Show("Column must be 0 through " + MAX_COL);
                        }
                    }
                    else
                    {
                        // Error message for invalid row.
                        MessageBox.Show("Column must be 0 through " + MAX_ROW);
                    }
                }
```

```csharp
            else
            {
                // Display an error message for noninteger column.
                MessageBox.Show("Enter an integer for the column.");
            }
        }
        else
        {
            // Display an error message for noninteger row.
            MessageBox.Show("Enter an integer for the row.");
        }
    }

    private void exitButton_Click(object sender, EventArgs e)
    {
        // Close the form.
        this.Close();
    }
}
}
```