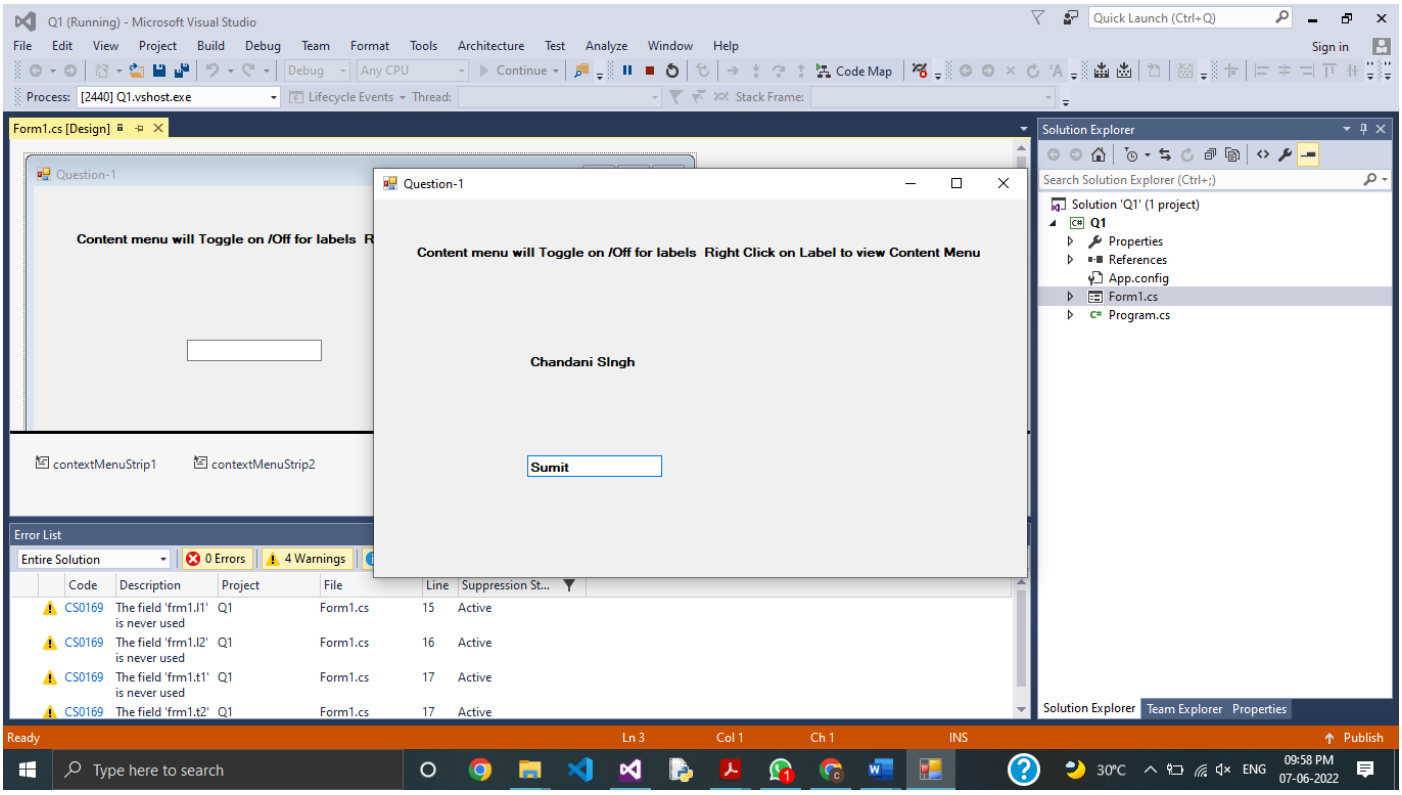
.Net Practical (207) – Assignment-2

/*

1) Add two labels to your form. Write code to toggle the labels on and off. The two labels should disappear with the textboxes. And they should reappear when the menu item is toggled to the on position

***/**



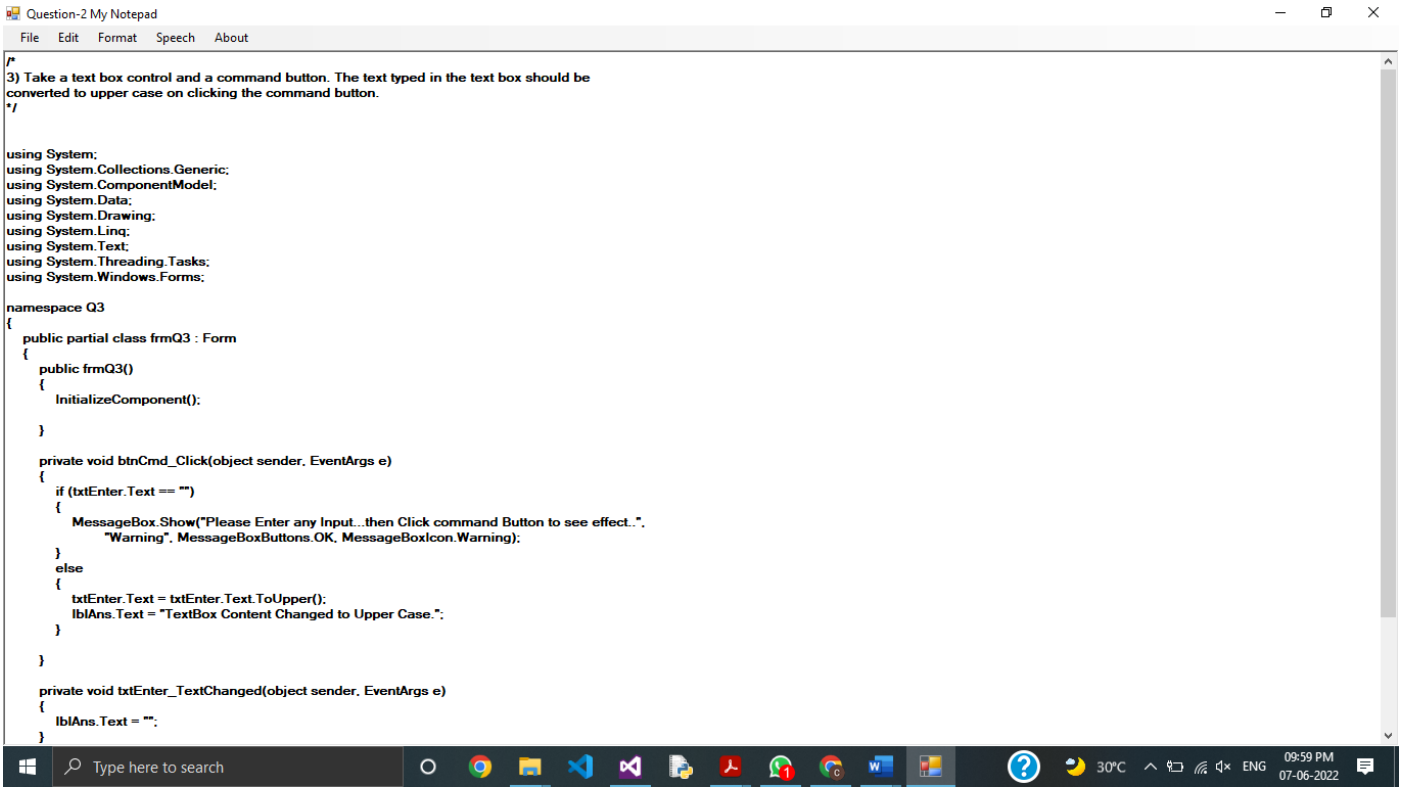
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Q1
{
    public partial class frm1 : Form
    {
        Label l1;
        Label l2;
        TextBox t1, t2;
        public frm1()
        {
            InitializeComponent();




/*
            l1 = new Label();
            l1.Height = 30;
            l1.Width = 70;
            l1.Left = 100;
            l1.Top = 50;
            l1.Text = "LABEL-1";
            ContextMenuStrip = contextMenuStrip1;
            l1.Visible = false;
            this.Controls.Add(l1);



            l2 = new Label();
            l2.Height = 30;
            l2.Width = 70;
            l2.Left = 100;
            l2.Top = 150;
            l2.Text = "LABEL-2";
            ContextMenuStrip = contextMenuStrip2;
            l2.Visible = false;
            this.Controls.Add(l2);

            t1 = new TextBox();
            t1.Height = 30;
            t1.Width = 70;
            t1.Left = 100;
            t1.Top = 50;
            ContextMenuStrip = contextMenuStrip1;
            t1.Visible = true;
            this.Controls.Add(t1);

            t2 = new TextBox();
            t2.Height = 30;
            t2.Width = 70;
```

```
            t2.Left = 100;
            t2.Top = 150;
            ContextMenuStrip = contextMenuStrip2;
            t2.Visible = true;
            this.Controls.Add(t2);
*/
        }

        private void onToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            if(textBox1.Text != "")
                label1.Text = textBox1.Text;
            label1.Visible = true;
            textBox1.Visible = false;

        }

        private void offToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            if(label1.Text != "Enter :")
                textBox1.Text = label1.Text;
            label1.Visible = false;
            textBox1.Visible = true;
            label2.Visible = true;
            textBox2.Visible = false;
        }
        private void onToolStripMenuItem2_Click(object sender, EventArgs e)
        {
            if (textBox2.Text != "")
                label2.Text = textBox2.Text;
            label2.Visible = true;
            textBox2.Visible = false;
        }

        private void offToolStripMenuItem2_Click(object sender, EventArgs e)
        {
            if (label2.Text != "Enter :")
                textBox1.Text = label1.Text;
            label2.Visible = false;
            textBox2.Visible = true;
            label1.Visible = true;
            textBox1.Visible = false;
        }
    }
}
```

/*

2) Create your own notepad editor with maximum functionalities.

*/



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Speech.Synthesis;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Q2
{
    public partial class frm2 : Form
    {
        String filename="";
        bool edited = true;

        public object PrintDialogBox { get; private set; }

        public frm2()
        {
            InitializeComponent();
            edited = false;
            filename = "";

        }

        private void newToolStripMenuItem_Click(object sender, EventArgs e)
        {
            if(edited == true)
            {
                var x = MessageBox.Show("Do you want to save", "Alert",
MessageBoxButtons.YesNoCancel,MessageBoxIcon.Warning);
                if(x.ToString() == "Yes")
                {
                    saveToolStripMenuItem_Click(sender,e);
                }
                else if(x.ToString() == "No")
                {
                    richTextBox1.Text = "";
                    edited = false;
                    filename = "";
                }
            }
        }

        private void openToolStripMenuItem_Click(object sender, EventArgs e)
        {
            /*
            openFileDialog1.Filter = "Text Files | *.txt";
            if (openFileDialog1.ShowDialog() == DialogResult.OK)
            {
                richTextBox1.Text = File.ReadAllText(openFileDialog1.FileName);
            }
            */

            if (edited == true)
```

```
            {
                var x = MessageBox.Show("Do you want to save", "Alert",
MessageBoxButtons.YesNoCancel, MessageBoxIcon.Warning);
                if (x.ToString() == "Yes")
                {
                    saveToolStripMenuItem_Click(sender, e);
                }
                else if (x.ToString() == "No")
                {
                    richTextBox1.Text = "";
                    edited = false;
                    filename = "";
                    //newToolStripMenuItem_Click(sender, e);
                  //  openFileDialog1.Filter = "Text Files | *.txt";
                    openFileDialog1.DefaultExt = "txt";
                    if (openFileDialog1.ShowDialog() == DialogResult.OK)
                    {
                        richTextBox1.Text =
File.ReadAllText(openFileDialog1.FileName);
                    }
                }
            }
            else
            {
                openFileDialog1.DefaultExt = "txt";
                if (openFileDialog1.ShowDialog() == DialogResult.OK)
                {
                    richTextBox1.Text = File.ReadAllText(openFileDialog1.FileName);
                }
            }


        }

        private void saveToolStripMenuItem_Click(object sender, EventArgs e)
        {
            /*
            String text = richTextBox1.Text;
            saveFileDialog1.DefaultExt = "txt";
            saveFileDialog1.ShowDialog();
            richTextBox1.SaveFile(saveFileDialog1.FileName);
            */

            SaveFileDialog saveFileDialog2 = new SaveFileDialog();

            saveFileDialog1.DefaultExt = "txt";

            switch (saveFileDialog2.ShowDialog())
            {
                case DialogResult.OK:
                    FileStream fileStream = new
FileStream(@saveFileDialog2.FileName,FileMode.Create,FileAccess.Write);
                    Byte[] text = Encoding.UTF8.GetBytes(richTextBox1.Text);
                    fileStream.Write(text, 0, richTextBox1.Text.Length);
                    fileStream.Close();
                    edited = false;
                    break;
                default:
                    break;
```

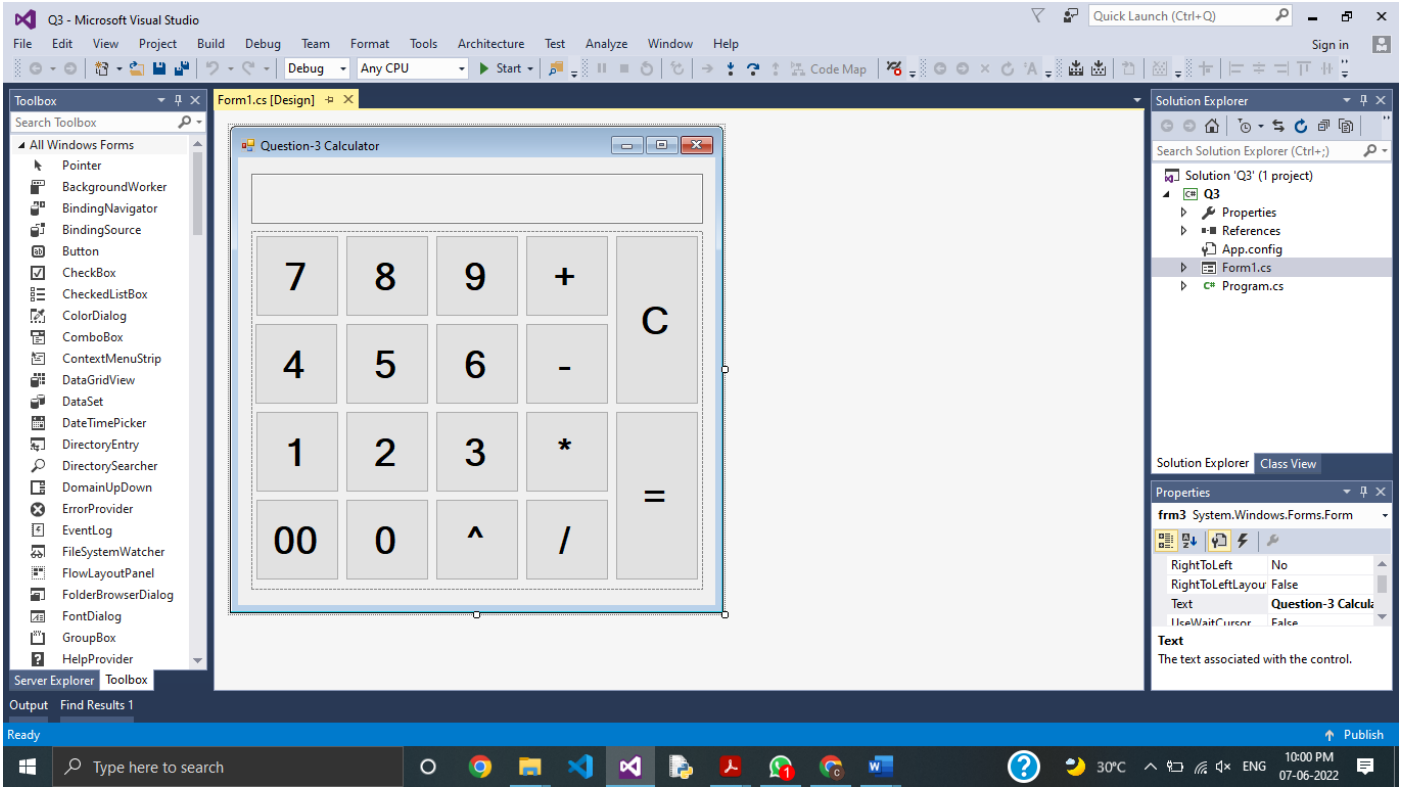
```
        }

    }

    private void saveAsToolStripMenuItem_Click(object sender, EventArgs e)
    {
        saveToolStripMenuItem_Click(sender, e);
    }

    private void printToolStripMenuItem_Click(object sender, EventArgs e)
    {
        PrintDialog printDialog1 = new PrintDialog();
        printDialog1.ShowDialog();
    }

    private void exitToolStripMenuItem_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }

    private void cutToolStripMenuItem_Click(object sender, EventArgs e)
    {
        richTextBox1.Cut();
    }

    private void copyToolStripMenuItem_Click(object sender, EventArgs e)
    {
        richTextBox1.Copy();
    }

    private void pasteToolStripMenuItem_Click(object sender, EventArgs e)
    {
        richTextBox1.Paste();
    }

    private void undoToolStripMenuItem_Click(object sender, EventArgs e)
    {
        richTextBox1.Undo();
    }

    private void redoToolStripMenuItem_Click(object sender, EventArgs e)
    {
        richTextBox1.Redo();
    }

    private void selectAllToolStripMenuItem_Click(object sender, EventArgs e)
    {
        richTextBox1.SelectAll();
    }

    private void fontToolStripMenuItem_Click(object sender, EventArgs e)
    {
        fontDialog1.ShowDialog();
        richTextBox1.SelectionFont = fontDialog1.Font;
    }

    private void colorToolStripMenuItem_Click(object sender, EventArgs e)
    {
        colorDialog1.ShowDialog();
```

```
            richTextBox1.SelectionColor = colorDialog1.Color;
        }

        private void maleVoiceToolStripMenuItem_Click(object sender, EventArgs e)
        {
            SpeechSynthesizer voice = new SpeechSynthesizer();
            voice.SelectVoiceByHints(VoiceGender.Male);
            voice.SpeakAsync(richTextBox1.Text);
        }

        private void femaleVoiceToolStripMenuItem_Click(object sender, EventArgs e)
        {
            SpeechSynthesizer voice = new SpeechSynthesizer();
            voice.SelectVoiceByHints(VoiceGender.Female);
            voice.SpeakAsync(richTextBox1.Text);
        }

        private void aboutToolStripMenuItem_Click(object sender, EventArgs e)
        {
            MessageBox.Show("It is My-Notepad");
        }

        private void richTextBox1_TextChanged(object sender, EventArgs e)
        {
            edited = true;
        }
    }
}
```

/*

3) Write a program to implement a simple calculator with memory and recall operations.

*/



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

namespace Q3
{
    public partial class frm3 : Form
    {
        public frm3()
        {
            InitializeComponent();
        }

        int ans=0;
        String op = "";
        bool num;

        private void num_click(object sender, EventArgs e)
        {
            Button btn = (Button)sender;
            if(txtEnter.Text != "0")
            {
                if(num == true)
                {
                    txtEnter.Text = txtEnter.Text + btn.Text;
                }
                else
                {
                    txtEnter.Text = btn.Text;
                    num = true;
                }

            }
        }

        private void btn7_Click(object sender, EventArgs e)
        {
            num_click(sender, e);
        }

        private void btn8_Click(object sender, EventArgs e)
        {
            num_click(sender, e);
        }

        private void btn9_Click(object sender, EventArgs e)
        {
            num_click(sender, e);
        }

        private void btn4_Click(object sender, EventArgs e)
        {
            num_click(sender, e);
```

```
}

private void btn5_Click(object sender, EventArgs e)
{
    num_click(sender, e);
}

private void btn6_Click(object sender, EventArgs e)
{
    num_click(sender, e);
}

private void btn1_Click(object sender, EventArgs e)
{
    num_click(sender, e);
}

private void btn2_Click(object sender, EventArgs e)
{
    num_click(sender, e);
}

private void btn3_Click(object sender, EventArgs e)
{
    num_click(sender, e);
}

private void btn00_Click(object sender, EventArgs e)
{
    num_click(sender, e);
}

private void btn0_Click(object sender, EventArgs e)
{
    num_click(sender, e);
}

private void btndot_Click(object sender, EventArgs e)
{
    num_click(sender, e);
}

private void btnadd_Click(object sender, EventArgs e)
{
    if (txtEnter.Text.Equals(""))
    {
        txtEnter.Text = "0";
    }
    if (ans == 0)
    {
        ans = Convert.ToInt32(txtEnter.Text);
    }
    else{
        ans = ans + Convert.ToInt32(txtEnter.Text);
    }
    txtEnter.Text = "";
    op = "+";
}
```

```
private void btnminus_Click(object sender, EventArgs e)
{
    if (txtEnter.Text.Equals(""))
    {
        txtEnter.Text = "0";
    }
    if (ans == 0)
    {
        ans = Convert.ToInt32(txtEnter.Text);
    }
    else
    {
        ans = ans - Convert.ToInt32(txtEnter.Text);
    }
    txtEnter.Text = "";
    op = "-";
}

private void btnmultiplication_Click(object sender, EventArgs e)
{
    if (txtEnter.Text.Equals(""))
    {
        txtEnter.Text = "0";
    }
    if (num == false)
    {
        ans = 1;
    }
    if (ans == 0)
    {
        ans = Convert.ToInt32(txtEnter.Text);
    }
    else
    {
        ans = ans * Convert.ToInt32(txtEnter.Text);
    }
    txtEnter.Text = "";
    op = "*";
}

private void btndivide_Click(object sender, EventArgs e)
{
    if (txtEnter.Text.Equals(""))
    {
        txtEnter.Text = "0";
    }
    if (ans == 0)
    {
        ans = Convert.ToInt32(txtEnter.Text);
    }
    else
    {
        ans = ans / Convert.ToInt32(txtEnter.Text);
    }
    txtEnter.Text = "";
    op = "/";
}

private void btnsquare_Click(object sender, EventArgs e)
```

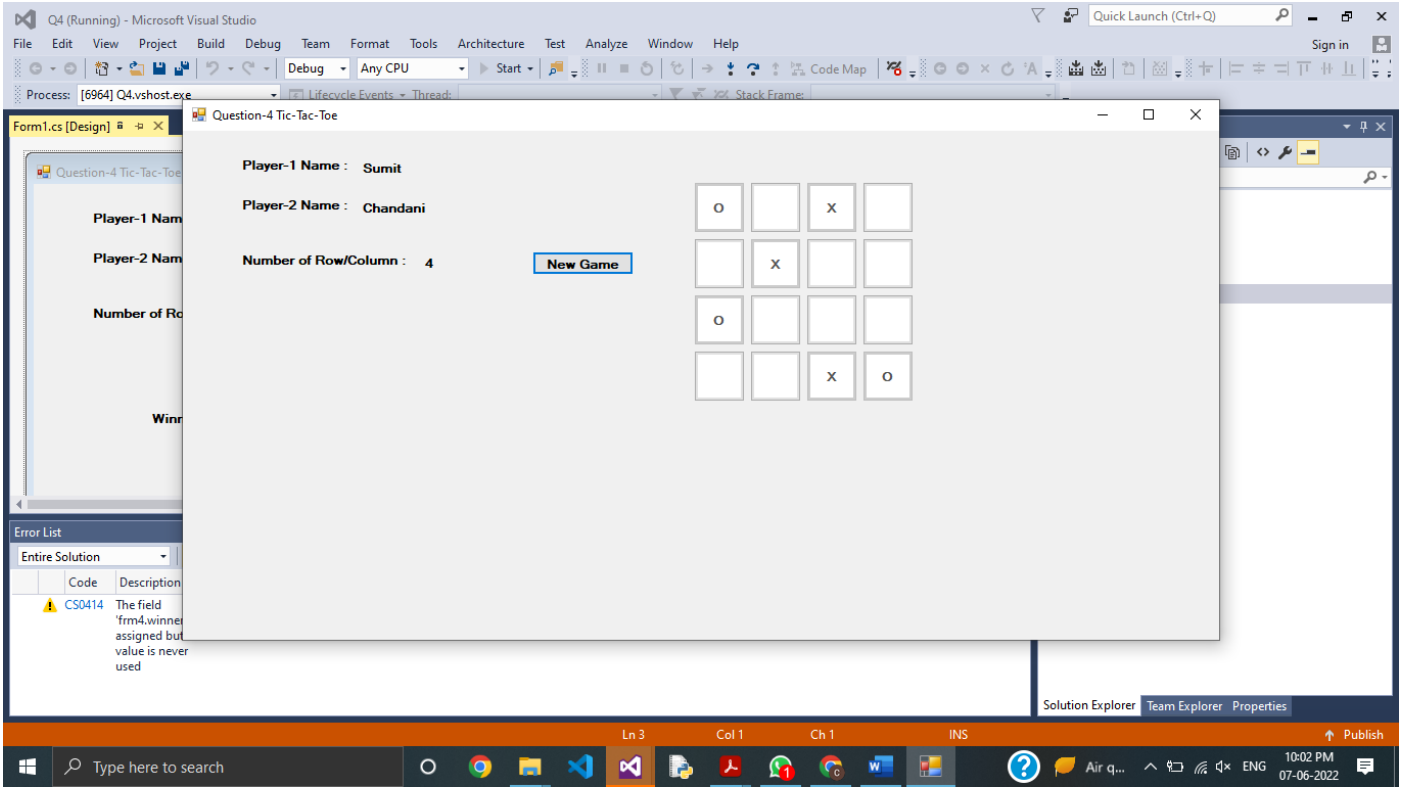
```
{
    if (txtEnter.Text.Equals(""))
    {
        txtEnter.Text = "0";
    }
    if (ans == 0)
    {
        ans = Convert.ToInt32(txtEnter.Text);
    }
    else
    {
        //ans = ans ^ Convert.ToInt32(txtEnter.Text);
        int temp = ans;
        for(int i=0;i< Convert.ToInt32(txtEnter.Text); i++)
        {
            ans = ans * temp;
        }

    }
    txtEnter.Text = "";
    op = "^";
}
private void btnC_Click(object sender, EventArgs e)
{
    ans = 0;
    txtEnter.Text = "";
    op = "";
}

private void btnans_Click(object sender, EventArgs e)
{
    if (txtEnter.Text.Equals(""))
    {
        txtEnter.Text = "0";
    }
    if (op.Equals("+")){
        ans = ans + Convert.ToInt32(txtEnter.Text);
    }
    else if (op.Equals("-")){
        ans = ans - Convert.ToInt32(txtEnter.Text);
    }
    else if (op.Equals("*")){
        ans = ans * Convert.ToInt32(txtEnter.Text);
    }
    else if (op.Equals("/")){
        ans = ans / Convert.ToInt32(txtEnter.Text);
    }
    else if (op.Equals("^"))
    {
        int temp = ans;
        for (int i = 1; i < Convert.ToInt32(txtEnter.Text); i++)
        {
            ans = ans * temp;
        }
    }



    txtEnter.Text = ans.ToString();
```

```
            // op = "";
             num = true;
        }


    }
}
```

/*

4) Write a program to develop a tic tac toe game with dynamic controls.

*/



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

namespace Q4
{
    public partial class frm4 : Form
    {
        public frm4()
        {
            InitializeComponent();
        }
        Button[,] btn;
        int n,count=0;
        bool turn= true; // true=X false=O
        String p1, p2;
        bool ans = true;
        String winner="";
        private void btnSubmit_Click(object sender, EventArgs e)
        {

            try
            {
                p1 = txtPlayer1.Text;
                p2 = txtPlayer2.Text;
                assign_name();
                create_button();
            }
            catch
            {
                MessageBox.Show("Enter No. of Rows..Valid...", "Warning",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
            }
            finally
            {

            }



        }

        private void btnClick(object sender,EventArgs e)
        {
            Button b = (Button)sender;
            count++;
            if (turn)
            {
                b.Text = "X";
            }
            else
            {
                b.Text = "O";
```

```
            }
            b.Enabled = false;
            bool result = checkWin();
            if (result == true)
            {
                if (turn)
                {
                    lblWinner.Text = p1;
                    MessageBox.Show("Winner : "+p1);
                }
                else
                {
                    lblWinner.Text = p2;
                    MessageBox.Show("Winner : "+p2);
                }
                lblWinner.Visible = true;
                for (int i = 0; i < n; i++)
                {
                    for (int j = 0; j < n; j++)
                    {
                        btn[i,j].Enabled = false;
                    }
                }
            }
            else if(count >= n*n)
            {
                lblWinner.Text = "Game Over";
                lblWinner.Visible = true;
                MessageBox.Show("Game Over...");
            }
            else
            {
                turn = !turn;
            }



        }

        private bool checkWin()
        {
            // for row check
            for (int i = 0; i < n; i++)
            {
                ans = true;
                for (int j = 1; j < n; j++)
                {
                    if ((Equals(btn[i, 0].Text.ToString(), btn[i, j].Text.ToString())
== true) && (btn[i, j].Enabled == false))
                    {
                        continue;
                    }
                    else
                    {
                        ans = false;
                    }
                }
                if(ans == true)
                {
```

```
                    return ans;
                }

            }

            // for column check
            for (int i = 0; i < n; i++)
            {
                ans = true;
                for (int j = 1; j < n; j++)
                {
                    if ((Equals(btn[0, i].Text.ToString(), btn[j, i].Text.ToString())
== true) && (btn[j, i].Enabled == false))
                    {
                        continue;
                    }
                    else
                    {
                        ans = false;
                    }
                }
                if (ans == true)
                {
                    return ans;
                }

            }

            // for first Diagonal check
            ans = true;
            for (int i = 1; i < n; i++)
            {
                if ((Equals(btn[0, 0].Text.ToString(), btn[i, i].Text.ToString()) ==
true) && (btn[i, i].Enabled == false))
                {
                    continue;
                }
                else
                {
                    ans = false;
                }
            }
            if (ans == true)
            {
                return ans;
            }

            // for second Diagonal check

            ans = true;
            for (int i = 1; i < n; i++)
            {
                int temp = n - i - 1;
                if ((Equals(btn[0, n-1].Text.ToString(), btn[i, temp].Text.ToString())
== true) && (btn[i,temp].Enabled == false))
                {
                    continue;
                }
                else
```

```
                {
                    ans = false;
                }
            }
            if (ans == true)
            {
                return ans;
            }

            return ans;

        }

        private void assign_name()
        {
            if(txtPlayer1.Text.Equals("") || txtPlayer2.Equals(""))
            {
                MessageBox.Show("Please enter Players
Name..","Warning",MessageBoxButtons.OK,MessageBoxIcon.Warning);
            }
            else if (txtNum.Text.Equals(""))
            {
                MessageBox.Show("Please enter No. of Row..", "Warning",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
            }
            else
            {
                n = Convert.ToInt32(txtNum.Text);


                lblPlayer1.Text = txtPlayer1.Text;
                lblPlayer2.Text = txtPlayer2.Text;
                lblNum.Text = txtNum.Text;

                txtPlayer1.Visible = false;
                txtPlayer2.Visible = false;
                txtNum.Visible = false;
                btnSubmit.Visible = false;

                lblPlayer1.Visible = true;
                lblPlayer2.Visible = true;
                lblNum.Visible = true;
                btnNewGame.Visible = true;

                lblWinner.Text = "";
                lblWinner.Visible = false;
                count = 0;

            }

        }

        private void btnNewGame_Click(object sender, EventArgs e)
        {
            turn = true;
            p1 = p2 = "";
            for(int i = 0; i < n; i++)
            {
                for(int j = 0; j < n; j++)
```

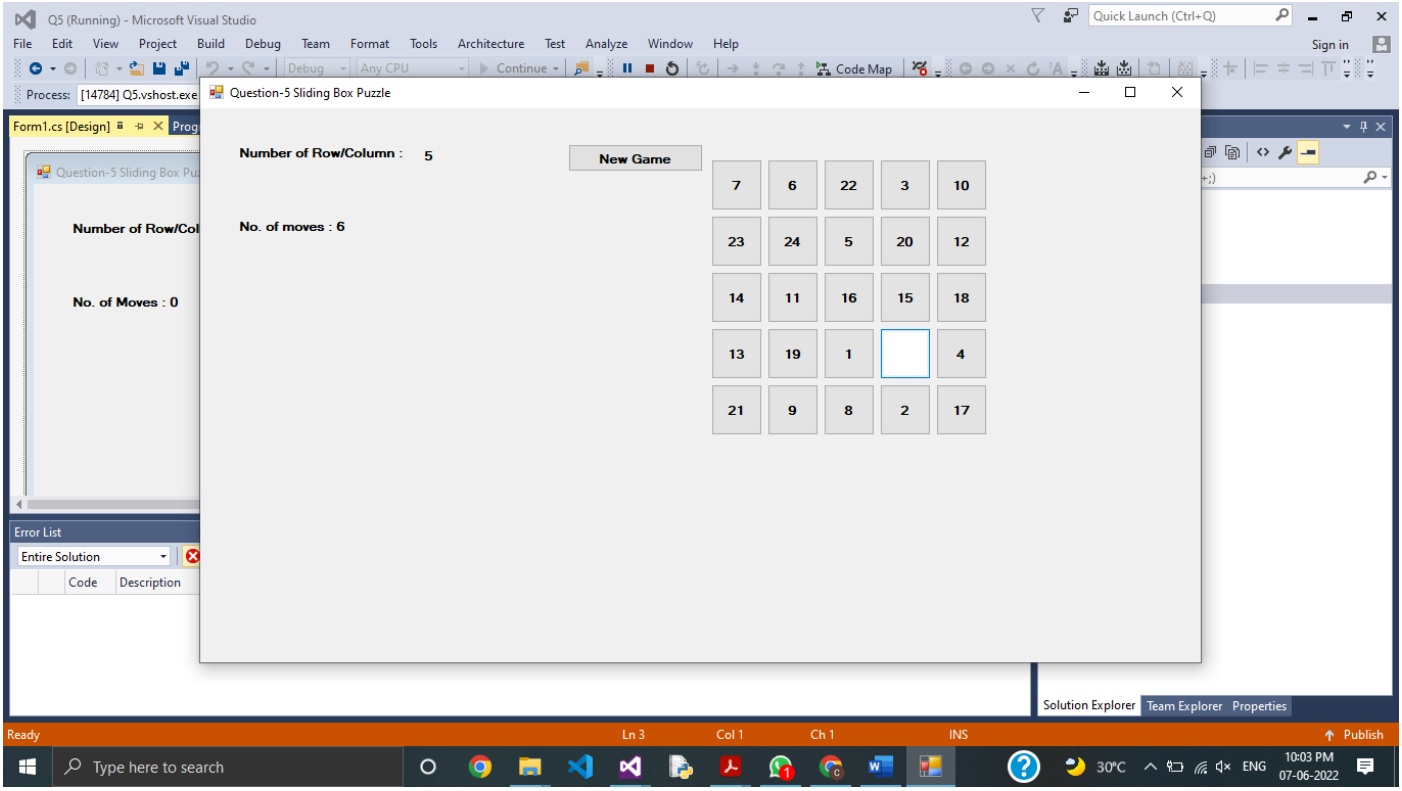
```
        {
            btn[i, j].Visible = false;
            btn[i, j] = null;
        }
    }

    btn = null;
    n = 0;
    lblPlayer1.Text = txtPlayer1.Text = "";
    lblPlayer2.Text = txtPlayer2.Text = "";
    lblNum.Text = txtNum.Text = "";

    txtPlayer1.Visible = true;
    txtPlayer2.Visible = true;
    txtNum.Visible = true;
    btnSubmit.Visible = true;

    lblPlayer1.Visible = false;
    lblPlayer2.Visible = false;
    lblNum.Visible = false;
    btnNewGame.Visible = false;

    lblWinner.Text = "";
    lblWinner.Visible = false;
    count = 0;
}

private void create_button()
{
    btn = new Button[n, n];
    int left = 500, top = 50;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            Button b = new Button();
            b.Height = 50;
            b.Width = 50;
            b.Left = left;
            b.Top = top;
            b.Text = "";
            b.BackColor = Color.White;
            b.Name = "btn" + i.ToString() + j.ToString();
            b.Visible = true;
            b.TabStop = false;
            b.Click += new EventHandler(btnClick);
            btn[i, j] = b;

            left += 55;
        }
        top += 55;
        left = 500;
    }
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            this.Controls.Add(btn[i, j]);
        }
```

```
            }
        }
    }
}
```

/*

5) Write a program to develop sliding box puzzle problem with dynamic controls.

*/



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

namespace Q5
{
    public partial class frm5 : Form
    {
        public frm5()
        {
            InitializeComponent();
        }

        int n;
        Button[] btn;
        int movesNumber = 0;


        private void btnSubmit_Click(object sender, EventArgs e)
        {
            try
            {
                assign_name();
                create_button();
            }
            catch
            {
                MessageBox.Show("Enter No. of Rows..Valid...", "Warning",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
            }
            finally
            {

            }
        }

        private void assign_name()
        {
            if (txtNum.Text.Equals(""))
            {
                MessageBox.Show("Please enter No. of Rows..", "Warning",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
            }
            else
            {
                n = Convert.ToInt32(txtNum.Text);

                lblNum.Text = txtNum.Text;

                txtNum.Visible = false;
                btnSubmit.Visible = false;

                lblNum.Visible = true;
```

```
            btnNewGame.Visible = true;

        }

    }

    private void create_button()
    {
        btn = new Button[n * n];
        int left = 500, top = 50;
        int temp = 0;
        int k = 0;
        List<int> labelList = new List<int>();
        Random rnd = new Random();
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < n; j++)
            {
                while (labelList.Contains(temp))
                {
                    temp = rnd.Next(n * n);
                }
                labelList.Add(temp);


                Button b = new Button();
                b.Height = 50;
                b.Width = 50;
                b.Left = left;
                b.Top = top;
                b.BackColor = Color.White;
                b.Visible = true;
                b.TabStop = false;
                b.Click += new EventHandler(btnClick);
                b.Name = (temp == 0) ? "" : temp.ToString();
                b.Text = (temp == 0) ? "" : temp.ToString();
                b.TabIndex = k;
                // b.Name = temp.ToString();
                //b.Text = b.Name;
                b.BackColor = (b.Text == "") ? Color.White :
Color.FromKnownColor(KnownColor.ControlLight);

                btn[k++] = b;
                //temp++;

                left += 55;
            }

            top += 55;
            left = 500;
        }
        for (int i = 0; i < n * n; i++)
        {
            this.Controls.Add(btn[i]);

        }
    }
```

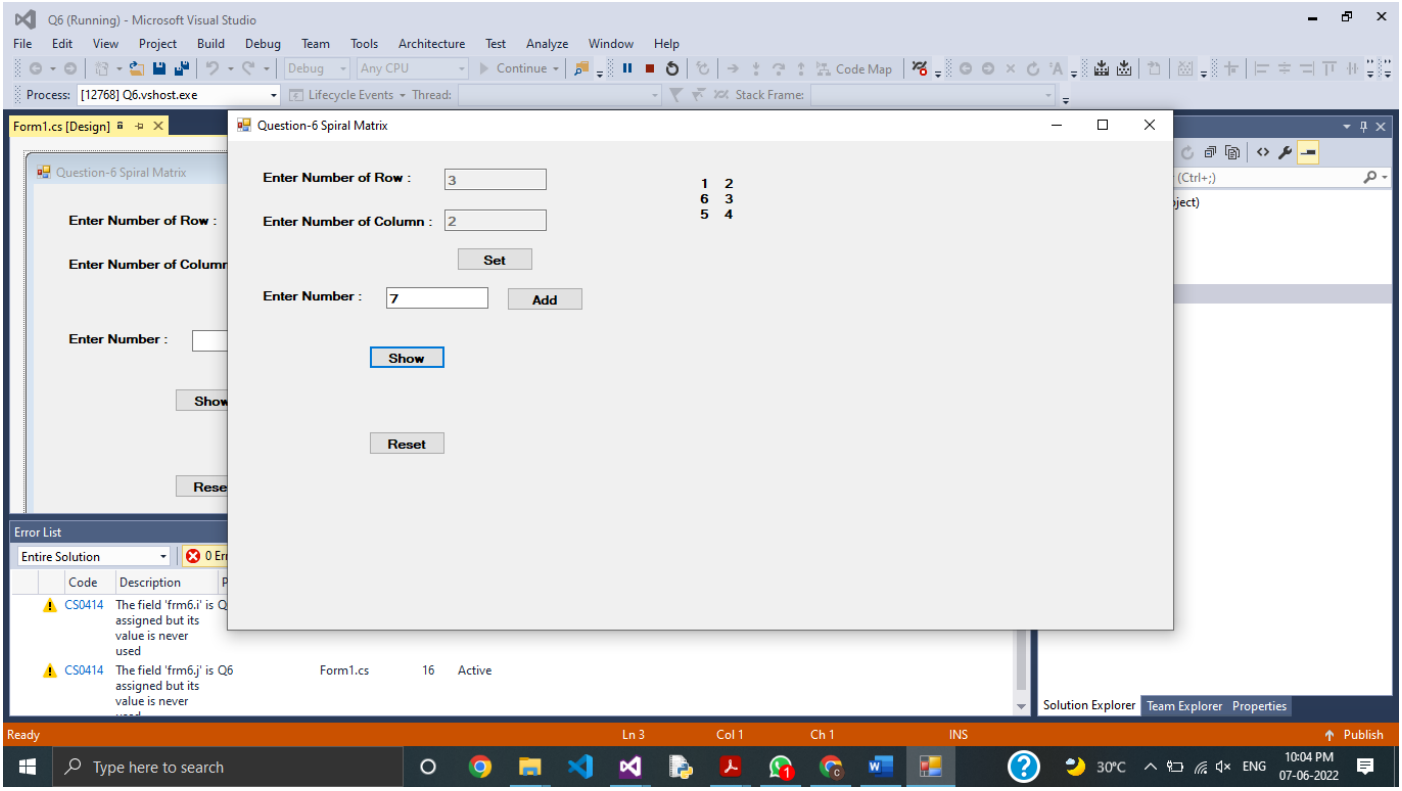
```
private void btnClick(object sender, EventArgs e)
{
    Button b = (Button)sender;
    if(b.Text == "")
    {
        return;
    }
    Button whiteBtn = null;
    foreach(Button bt in btn)
    {
        if(bt.Text == "")
        {
            whiteBtn = bt;
            break;
        }
    }
    /*
    if(Convert.ToInt32(b.Text) == (Convert.ToInt32(whiteBtn.Text) -1) ||
       Convert.ToInt32(b.Text) == (Convert.ToInt32(whiteBtn.Text) - n) ||
       Convert.ToInt32(b.Text) == (Convert.ToInt32(whiteBtn.Text) + 1) ||
       Convert.ToInt32(b.Text) == (Convert.ToInt32(whiteBtn.Text) + n))
        */
    if (b.TabIndex == (whiteBtn.TabIndex  - 1) ||
        b.TabIndex == (whiteBtn.TabIndex - n) ||
        b.TabIndex == (whiteBtn.TabIndex + 1) ||
        b.TabIndex == (whiteBtn.TabIndex + n))
    {
        whiteBtn.BackColor = Color.FromKnownColor(KnownColor.ControlLight);
        b.BackColor = Color.White;
        whiteBtn.Text = b.Text;
        b.Text = "";
        movesNumber++;
        lblNoOfMoves.Text = "No. of moves : " + movesNumber.ToString();
    }
    // int cur = Convert.ToInt32(b.Name);
    // b.BackColor = Color.Aqua;

    checkOrder();
}

private void checkOrder()
{
    int index = 1;
    foreach(Button bt in btn)
    {
        if(bt.Text!="" && Convert.ToInt32(bt.Text) != index)
        {
            return;
        }
        index++;
    }
    MessageBox.Show("Congrats ! You did it in " + movesNumber + " moves");
}
private void btnNewGame_Click(object sender, EventArgs e)
{

    for (int i = 0; i < n*n; i++)
    {
        btn[i].Visible = false;
```

```
                btn[i] = null;

            }

            btn = null;
            n = 0;
            lblNum.Text = txtNum.Text = "";

            txtNum.Visible = true;
            btnSubmit.Visible = true;

            lblNum.Visible = false;
            btnNewGame.Visible = false;

            lblNoOfMoves.Text = "No. of moves : 0";
            movesNumber = 0;

        }




    }
}
```

/*

6) Write a program to create circular matrix

*/



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

namespace Q6
{
    public partial class frm6 : Form
    {
        int r = 0, c = 0;
        int i, j;
        int rs, re, cs, ce;
        int[] arr;
        int[,] newarr;
        int l = 0;
        public frm6()
        {
            InitializeComponent();
            i = 0;
            j = -1;
        }

        private void btnSet_Click(object sender, EventArgs e)
        {
            try
            {
                r = Convert.ToInt32(txtRow.Text);
                if (r <= 0)
                {
                    r = 0;
                }
            }
            catch
            {
                MessageBox.Show("Please Enter valid Number of Row..", "Warning",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
            }
            try
            {
                c = Convert.ToInt32(txtCol.Text);
                if (c <= 0)
                {
                    c = 0;
                }
            }
            catch
            {
                MessageBox.Show("Please Enter valid Number of Column..", "Warning",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
            }
            if (r > 0 && c > 0)
            {
                arr = new int[r * c];
                newarr = new int[r, c];
```

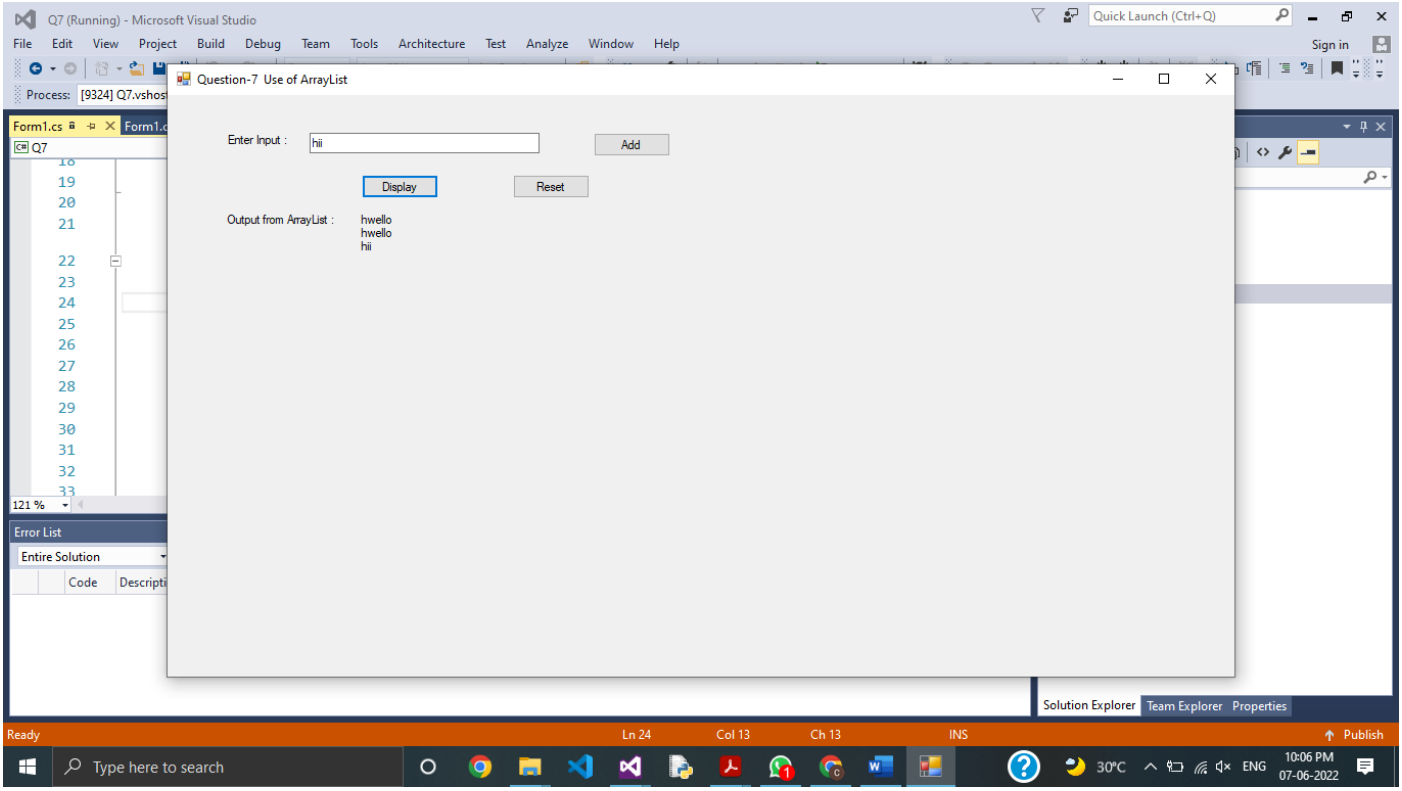
```
            txtRow.Enabled = false;
            txtCol.Enabled = false;
        }
    }



    private void btnAdd_Click(object sender, EventArgs e)
    {
        if(r>0 && c > 0)
        {


            if (l >= r*c)
            {
                MessageBox.Show("Matrix is full...", "Warning");
            }
            else
            {
                int num = Convert.ToInt32(txtNum.Text);
                arr[l++] = num;
            }


        }
        else
        {
            MessageBox.Show("Please Enter value for Row and Column.", "Warning",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
    }

    private void btnShow_Click(object sender, EventArgs e)
    {

        rs = 0;
        cs = 0;
        re = r;
        ce = c;
        int k = 0;
        while (rs < re && cs < ce && k < r * c)
        {
            /* first row from remaining rows */
            for (int i = cs; i < ce; i++)
            {
                newarr[rs, i] = arr[k++];
            }
            rs++;

            /* last column from reamining columns */
            for (int i = rs; i < re; i++)
            {
                newarr[i, ce - 1] = arr[k++];
            }
            ce--;

            /* last row from reming rows */
            if (rs < re)
            {
```

```
                    for (int i = ce - 1; i >= cs; i--)
                    {
                        newarr[re - 1, i] = arr[k++];
                    }
                    re--;
                }
                if (cs < c)
                {
                    for (int i = re - 1; i >= rs; i--)
                    {
                        newarr[i, cs] = arr[k++];
                    }
                    cs++;
                }

            }


            if (r > 0 && c > 0)
            {
                txtDisplay.Text = "";
                for (int i = 0; i < r; i++)
                {
                    for (int j = 0; j < c; j++)
                    {
                        txtDisplay.Text = txtDisplay.Text + newarr[i, j].ToString() +
"    ";
                    }
                    txtDisplay.Text = txtDisplay.Text + "\n";
                }
            }
            else
            {
                MessageBox.Show("Please Enter value for Row and Column.", "Warning",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
            }
        }

        private void btnReset_Click(object sender, EventArgs e)
        {
            r = 0; c = 0; i = 0; j = -1;
            arr = null;
            newarr = null;

            txtRow.Text = "";
            txtCol.Text = "";
            txtNum.Text = "";
            txtRow.Enabled = true;
            txtCol.Enabled = true;
            txtNum.Enabled = true;
            txtDisplay.Text = "";
        }

    }
}
```

/*

7) Write a program to demonstrate use of array list.

*/

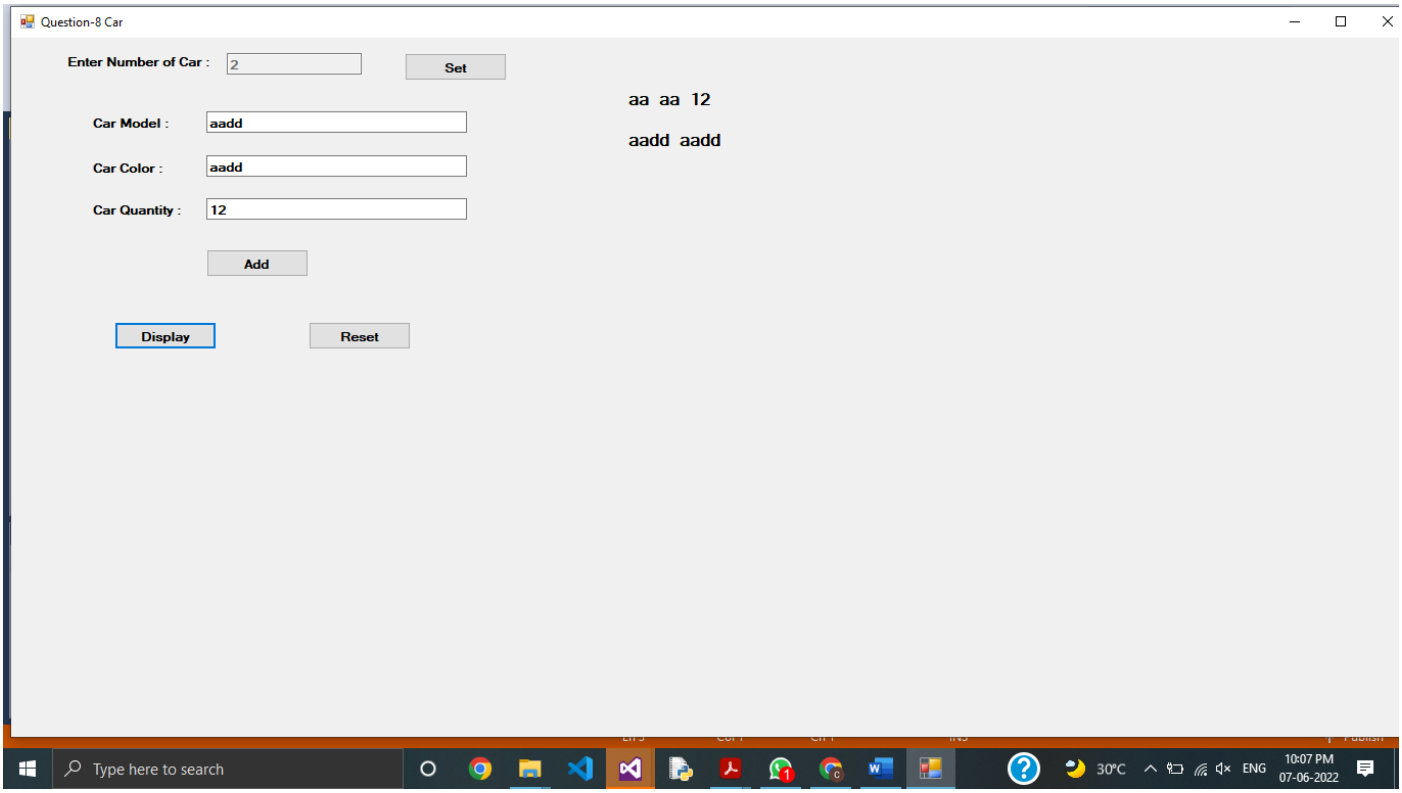


```
using System;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Q7
{
    public partial class frm7 : Form
    {
        public frm7()
        {
            InitializeComponent();
        }

        ArrayList a = new ArrayList();
        private void btnAdd_Click(object sender, EventArgs e)
        {
            try
            {
                String input = txtEnter.Text.ToString();
                a.Add(input);
            }
            catch
            {
                MessageBox.Show("Please Enter Valid String..");
            }
        }

        private void btnDisplay_Click(object sender, EventArgs e)
        {
            foreach(String i in a)
            {
                lblList.Text += i.ToString()+ "\n";
            }
        }

        private void btnReset_Click(object sender, EventArgs e)
        {
            a = null;
            lblList.Text = "";
            txtEnter.Text = "";
        }
    }
}
```

/*

8) Create a class car with essential properties and methods, create UI to take input of the details of n number of cars and display it using dynamic controls.

*/



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

namespace Q8
{
    public partial class frm8 : Form
    {
        public frm8()
        {
            InitializeComponent();
        }

        int n=0,i=0;
        Label[] lbl;

        class car
        {
            private String model;
            private String color;
            private int qty;
            public String msg;

            public car(String model,String color,int qty)
            {
                this.model = model;
                this.color = color;
                this.qty = qty;

                this.msg = this.model + "   " + this.color + "   " +
this.qty.ToString();
               // MessageBox.Show(this.msg);

            }
        }

        car[] c;
        private void btnSet_Click(object sender, EventArgs e)
        {
            try
            {

                int temp = Convert.ToInt32(txtNum.Text);
                if (temp <= 0) {
                    MessageBox.Show("Please Enter Positive Number of Car.. ");
                }
                else if(temp > 50){
                    MessageBox.Show("Plese ENter Car number between 1 to 50");
                }
                else
                {
                    n = temp;
                    c = new car[n];
```

```
            txtNum.Enabled = false;
        }
    }
    catch
    {
        MessageBox.Show("Please Enter Valid Number of Car..");
    }

}


private void btnAdd_Click(object sender, EventArgs e)
{
    if(i < n)
    {
        try
        {
            String m = txtModel.Text;
            String c1 = txtColor.Text;
            int q = Convert.ToInt32(txtQty.Text);
            c[i] = new car(m, c1, q);
            i++;
            MessageBox.Show("Added Successfully..");
        }
        catch
        {
            MessageBox.Show("Please Enter Valid Info..");
        }

    }
    else
    {
        MessageBox.Show("Array Full..");
    }

}

private void btnReset_Click(object sender, EventArgs e)
{
    lbl = null;
    c = null;
    i = 0;
    n = 0;
}

private void btnDisplay_Click(object sender, EventArgs e)
{
    lbl = new Label[i];
    int l = 600, t = 50;


    for(int k = 0; k < i; k++)
    {
        Label lb = new Label();
        lb.Left = l;
        lb.Top = t;
        lb.Font = new Font("Microsoft Sans Serif", 12, FontStyle.Bold);

        t = t + 40;
```

```
                lb.Name = "lbl_"+ k.ToString();
                lb.Text = c[k].msg;

                lbl[k] = lb;
            }

            for(int k = 0; k < i; k++)
            {
                this.Controls.Add(lbl[k]);
            }


        }

    }
}
```