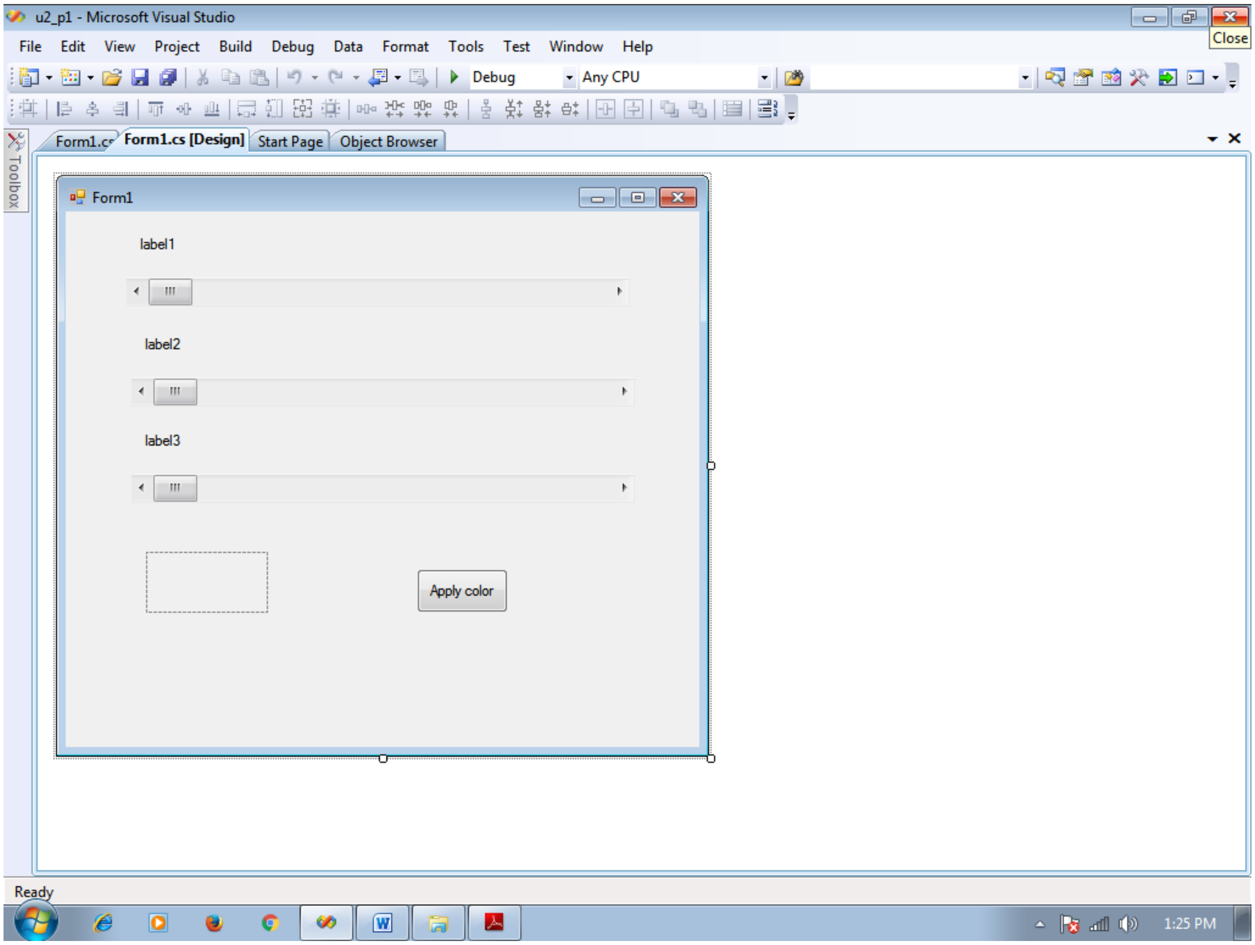
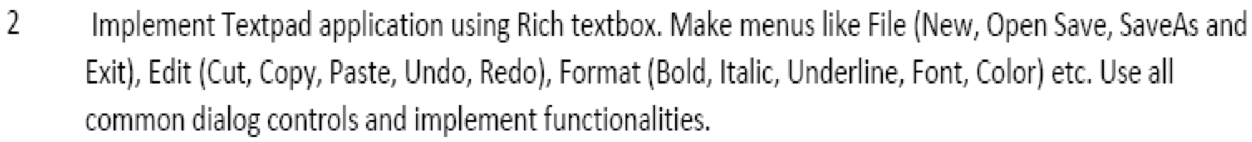
1 Take 3 Scrollbars indicating values of Red, Green and Blue colors from 0 to 255. Take 3 labels to show the values of the scroll bars. Depending upon values selected by the user with the help of scrollbars make a color and shows it in the picture box. Place a button having text 'Apply Color', and on the click event of the button apply that color to the form.

**(toolbox-3 hscrollbar ,1 picturebox,1 button)**



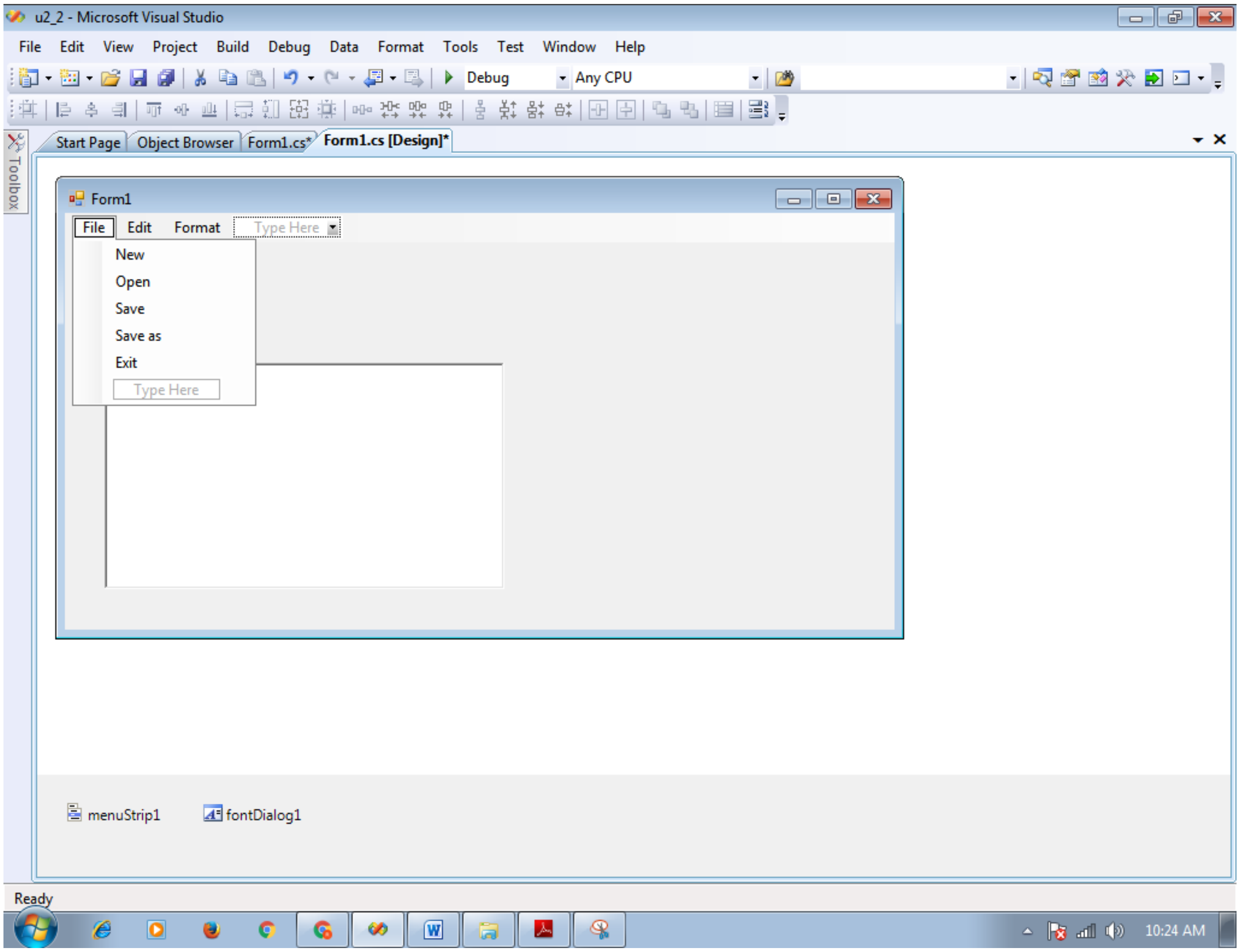
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace u2_p1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void label1_Click(object sender, EventArgs e)
        {

        }

        private void hScrollBar2_Scroll(object sender, ScrollEventArgs e)
        {
            pictureBox1.BackColor = Color.FromArgb(hScrollBar1.Value,
hScrollBar2.Value, hScrollBar3.Value);
            label2.Text = hScrollBar2.Value.ToString();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            pictureBox1.BorderStyle = BorderStyle.Fixed3D;
        }

        private void hScrollBar1_Scroll(object sender, ScrollEventArgs e)
        {
            pictureBox1.BackColor =
Color.FromArgb(hScrollBar1.Value,hScrollBar2.Value,hScrollBar3.Value);
            label1.Text = hScrollBar1.Value.ToString();
        }

        private void hScrollBar3_Scroll(object sender, ScrollEventArgs e)
        {
            pictureBox1.BackColor = Color.FromArgb(hScrollBar1.Value,
hScrollBar2.Value, hScrollBar3.Value);
            label3.Text = hScrollBar3.Value.ToString();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.BackColor = pictureBox1.BackColor;
        }
    }
}
```

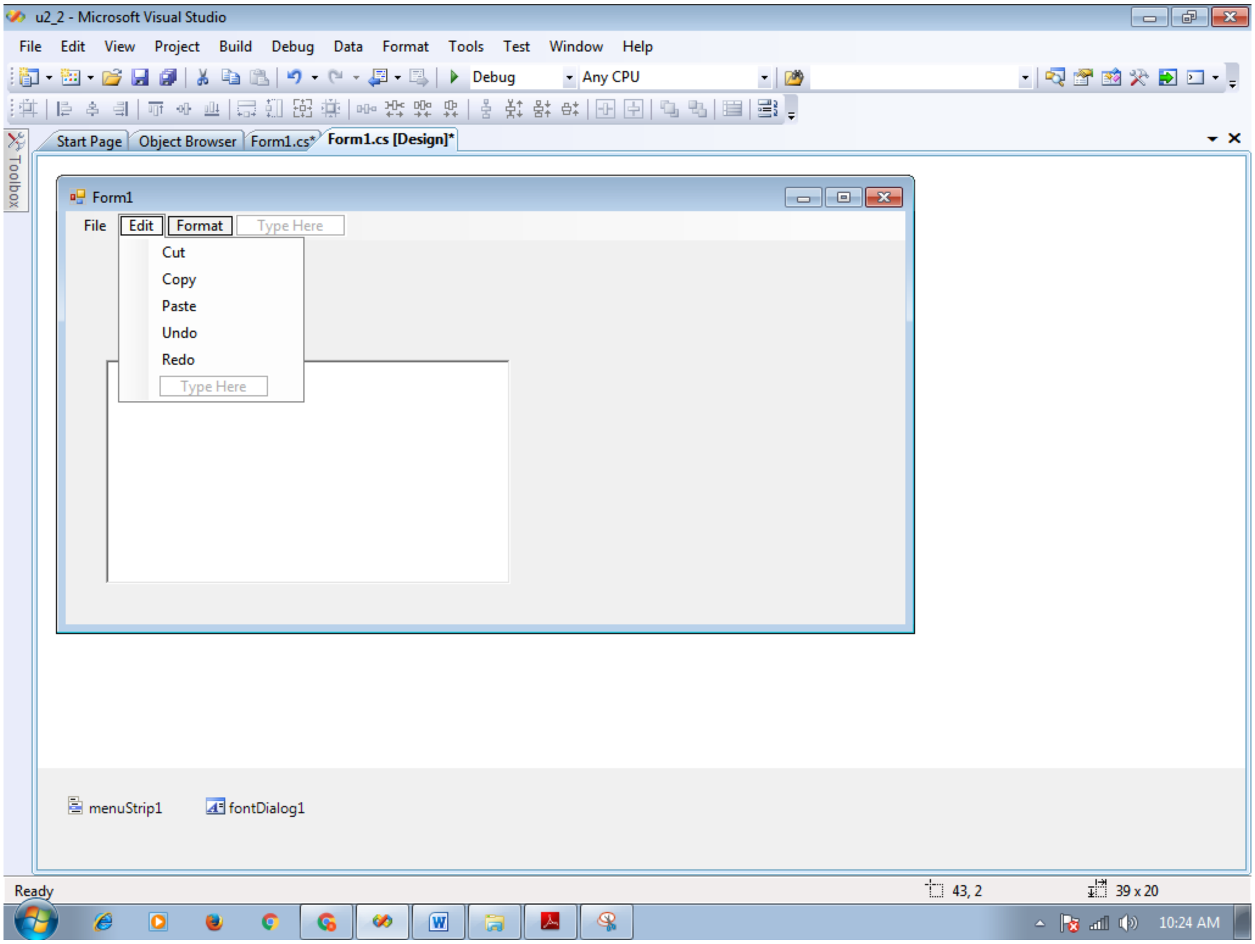


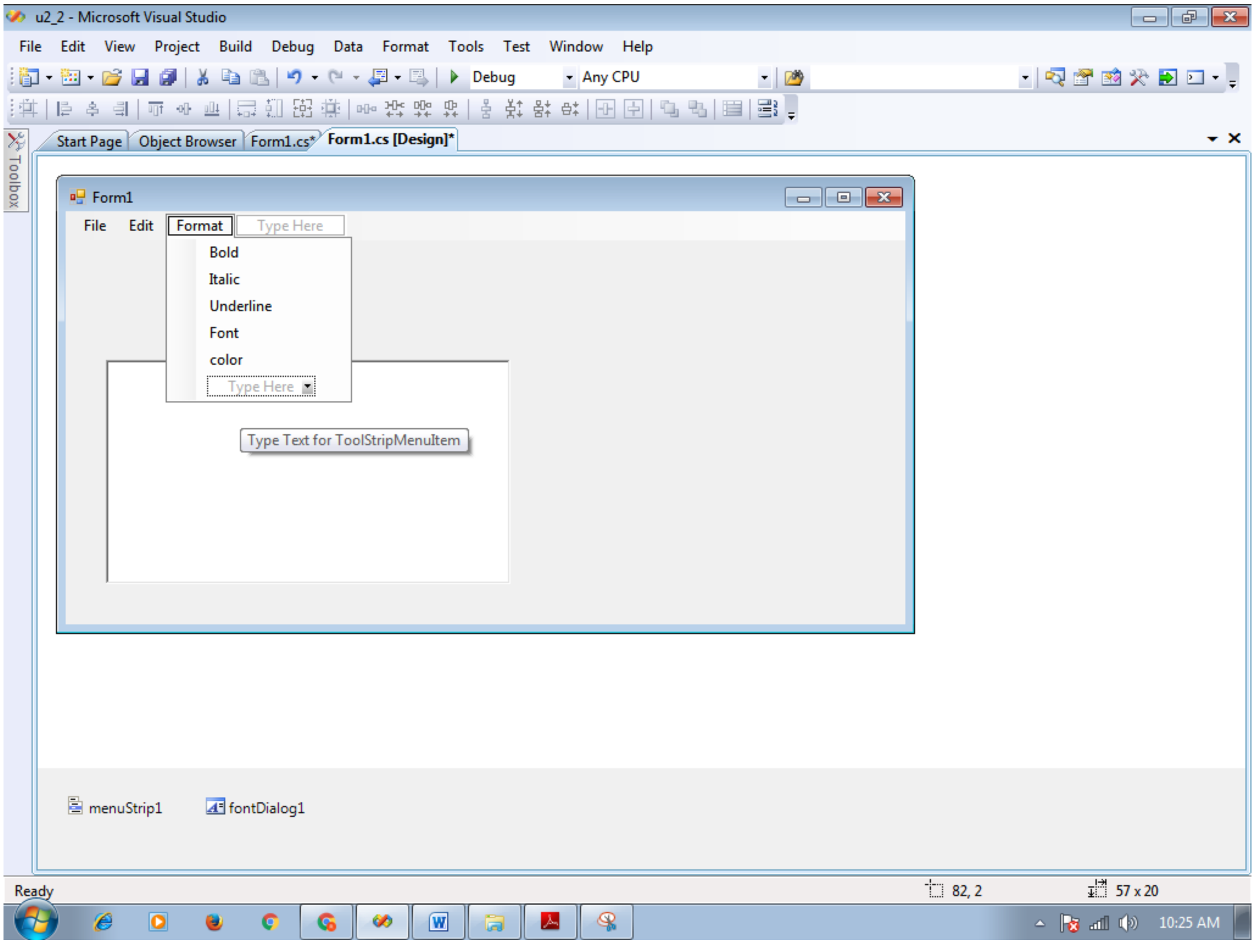
2 Implement Textpad application using Rich textbox. Make menus like File (New, Open Save, SaveAs and Exit), Edit (Cut, Copy, Paste, Undo, Redo), Format (Bold, Italic, Underline, Font, Color) etc. Use all common dialog controls and implement functionalities.

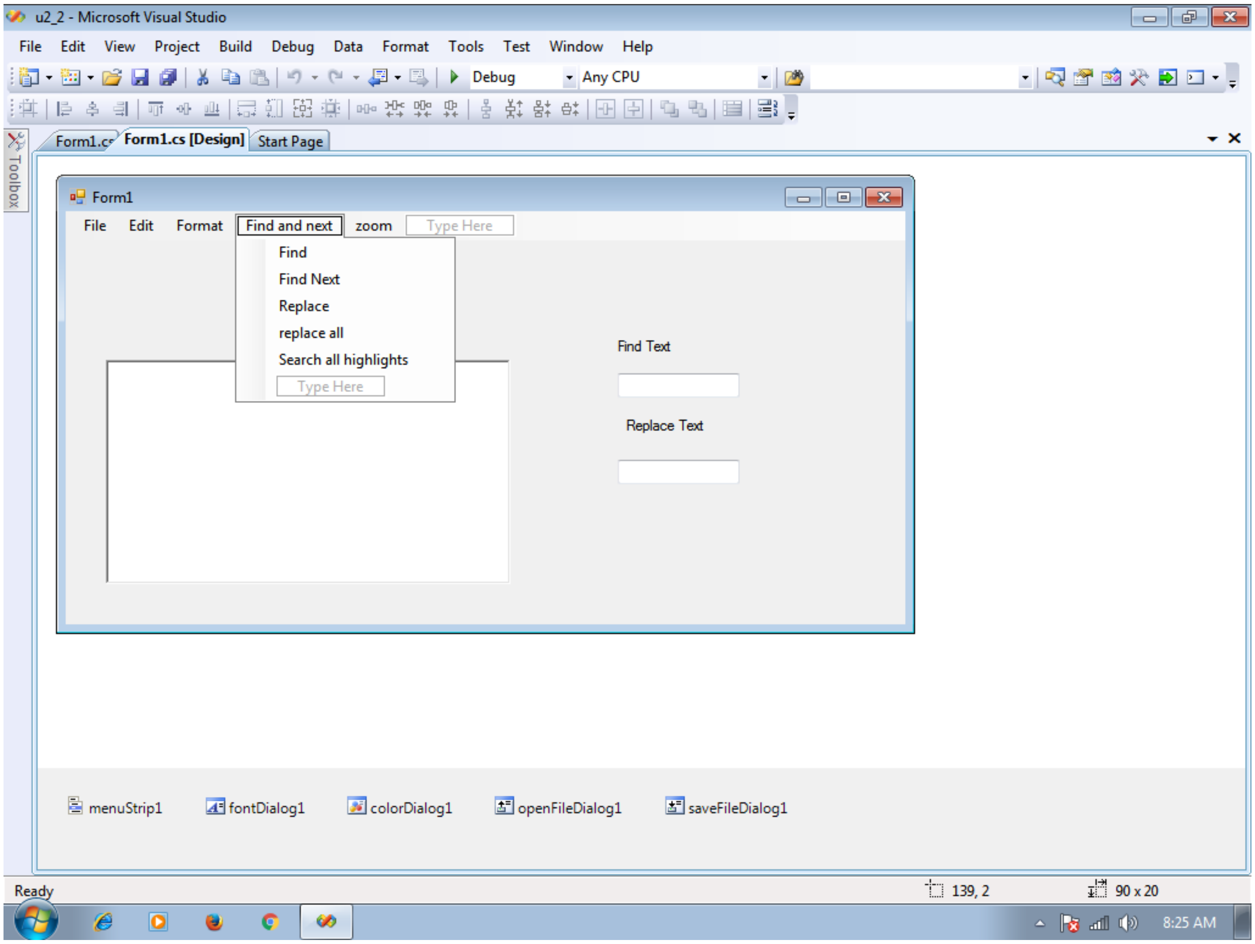
**3. Take a Ritch Textbox and implements find, find next, replace and replace all functionalities.**

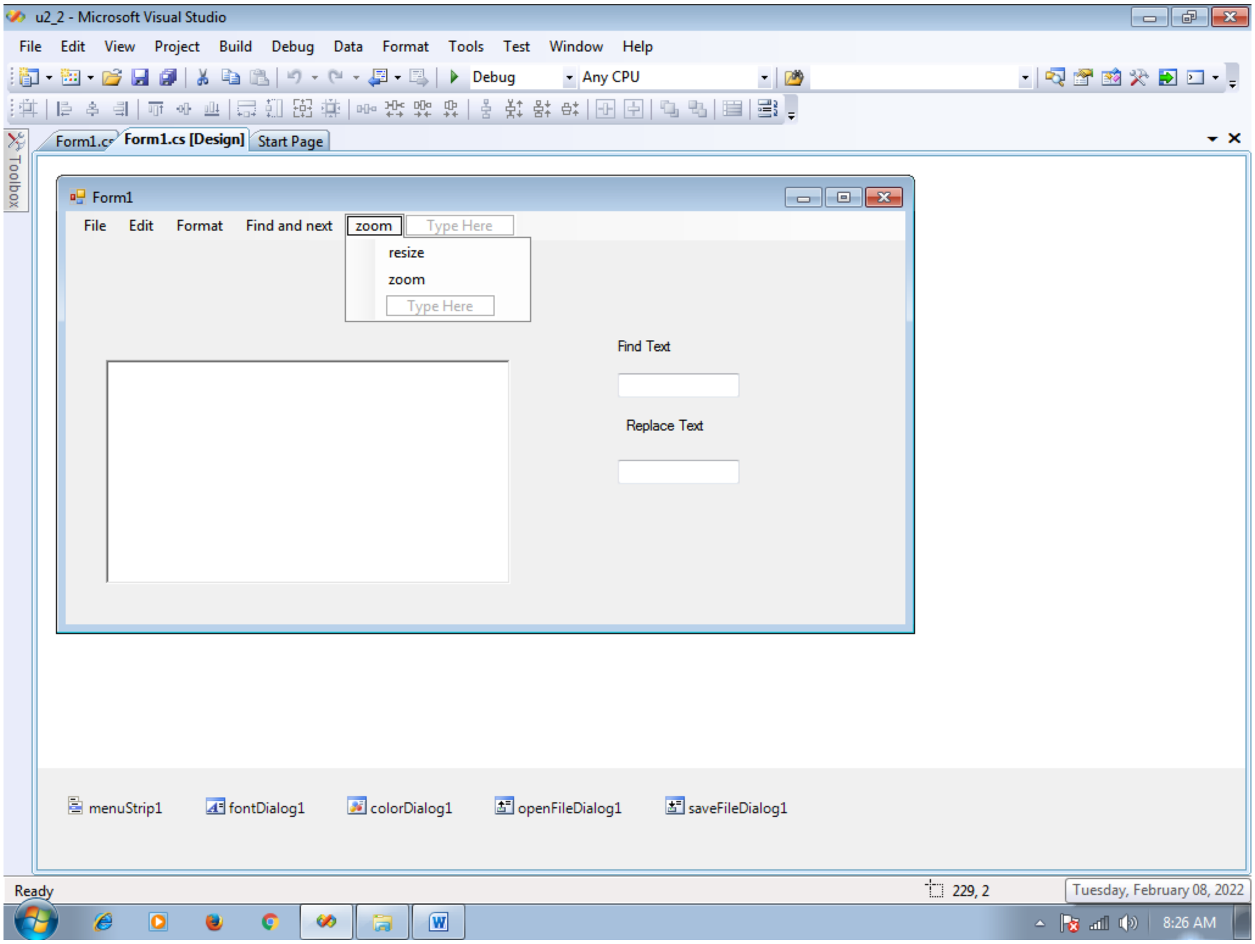
**(toolbox- 1ritch textbox,1 menustrip,fontdialog,savefiledialog,colordialog,openfiledialog,2 textbox,2 label)**


u2_2 - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Test Window Help
Debug
Any CPU
Start Page
Object Browser
Form1.cs*
Form1.cs [Design]*
Toolbox
Form1
File
Edit
Format
Type Here
New
Open
Save
Save as
Exit
Type Here
menuStrip1
fontDialog1
Ready
10:24 AM


u2_2 - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Test Window Help
Debug
Any CPU
Toolbox
Start Page
Object Browser
Form1.cs*
Form1.cs [Design]*
Form1
File
Edit
Format
Type Here
Cut
Copy
Paste
Undo
Redo
Type Here
menuStrip1
fontDialog1
Ready
43, 2
39 x 20
10:24 AM


u2_2 - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Test Window Help
Debug
Any CPU
Start Page
Object Browser
Form1.cs*
Form1.cs [Design]*
Toolbox
Form1
File
Edit
Format
Type Here
Bold
Italic
Underline
Font
color
Type Here
Type Text for ToolStripMenuItem
menuStrip1
fontDialog1
Ready
82, 2
57 x 20
10:25 AM


u2_2 - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Test Window Help
Debug
Any CPU
Toolbox
Form1.cs
Form1.cs [Design]
Start Page
Form1
File Edit Format Find and next zoom Type Here
Find
Find Next
Replace
replace all
Search all highlights
Type Here
Find Text
Replace Text
menuStrip1
fontDialog1
colorDialog1
openFileDialog1
saveFileDialog1
Ready
139, 2
90 x 20
8:25 AM



code

```
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 u2_2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void newToolStripMenuItem_Click(object sender, EventArgs e)
        {
            richTextBox1.Clear();
        }
        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 boldToolStripMenuItem_Click(object sender, EventArgs e)
        {
            richTextBox1.SelectionFont = new Font(richTextBox1.SelectionFont,
FontStyle.Bold);
        }

        private void italicToolStripMenuItem_Click(object sender, EventArgs
e)
        {
            richTextBox1.SelectionFont = new Font(richTextBox1.SelectionFont,
FontStyle.Italic);
        }

        private void underlineToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            richTextBox1.SelectionFont = new Font(richTextBox1.SelectionFont,
FontStyle.Underline);
        }

        private void fontToolStripMenuItem_Click(object sender, EventArgs e)
        {

if(fontDialog1.ShowDialog()==System.Windows.Forms.DialogResult.OK)

            {
                richTextBox1.SelectionFont = fontDialog1.Font;
            }
        }
        private void colorToolStripMenuItem_Click(object sender, EventArgs e)
        {
            if (colorDialog1.ShowDialog() ==
System.Windows.Forms.DialogResult.OK)
            {
                richTextBox1.SelectionColor = colorDialog1.Color;
            }
        }

        private void openToolStripMenuItem_Click(object sender, EventArgs e)
        {
            openFileDialog1.Filter = "TXT|*.txt|DOC|*.doc";
            if (openFileDialog1.ShowDialog() ==
System.Windows.Forms.DialogResult.OK)
            {
                richTextBox1.LoadFile(openFileDialog1.FileName,
RichTextBoxStreamType.PlainText);
                this.Text = openFileDialog1.FileName;
            }
        }

        private void saveToolStripMenuItem_Click(object sender, EventArgs e)
        {
            saveFileDialog1.Filter = "TXT|*.txt|DOC|*.doc";
            if (saveFileDialog1.ShowDialog() ==
System.Windows.Forms.DialogResult.OK)
            {
                richTextBox1.LoadFile(saveFileDialog1.FileName,
RichTextBoxStreamType.PlainText);
            }
        }

        private void saveAsToolStripMenuItem_Click(object sender, EventArgs
e)
        {
            richTextBox1.SaveFile(this.Text,
RichTextBoxStreamType.PlainText);
        }

        private void exitToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Close();

        }


        private void findToolStripMenuItem_Click(object sender, EventArgs e)
        {
            int pos;
            pos = richTextBox1.Find(textBox1.Text, 1,
richTextBox1.TextLength, RichTextBoxFinds.None);
            if (pos > 0)
            {
                richTextBox1.SelectionStart = pos;
                richTextBox1.SelectionLength = textBox1.TextLength;
                richTextBox1.SelectionColor = Color.Blue;
            }
        }
        private void replaceToolStripMenuItem_Click(object sender, EventArgs
e)
        {
            richTextBox1.SelectedText = textBox2.Text;
        }

        private void findNextToolStripMenuItem_Click(object sender, EventArgs
e)
        {
            int pos;
            pos = richTextBox1.Find(textBox1.Text, 1,
richTextBox1.TextLength, RichTextBoxFinds.None);
            if (pos > 0)
            {
                richTextBox1.SelectionStart = pos;
                richTextBox1.SelectionLength = textBox1.TextLength;
                richTextBox1.SelectionColor = Color.Blue;
            }
        }

        private void replacAllToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            richTextBox1.Text = richTextBox1.Text.Replace(textBox1.Text,
textBox2.Text);
        }

        private void zoomToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            richTextBox1.ZoomFactor = 5;
        }

        private void resizeToolStripMenuItem_Click(object sender, EventArgs
e)
        {
            richTextBox1.Height = this.Height;
            richTextBox1.Width = this.Width;
        }

        private void searchAllHighlightsToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            int index = 0;
            Int64 startSearch = 0;
            index = richTextBox1.Find(textBox1.Text, 0);
            while (index > -1)
            {
                index = richTextBox1.Find(textBox1.Text,
Convert.ToInt16(startSearch), richTextBox1.TextLength,
RichTextBoxFinds.None);
                richTextBox1.SelectionColor = Color.Red;
                startSearch = index + 1;
            }
        }

          }
}
```

The **StreamReader** and **StreamWriter** classes are used for reading from and writing data to text files. These classes inherit from the abstract base class Stream, which supports reading and writing bytes into a file stream.

## The StreamReader Class

The **StreamReader** class also inherits from the abstract base class TextReader that represents a reader for reading series of characters. The following table describes some of the commonly used **methods** of the StreamReader class −

| Sr.No. | Method & Description |
|---|---|
| 1 | **public override void Close()**<br>It closes the StreamReader object and the underlying stream, and releases any system resources associated with the reader. |
| 2 | **public override int Peek()**<br>Returns the next available character but does not consume it. |
| 3 | **public override int Read()**<br>Reads the next character from the input stream and advances the character position by one. |

## The StreamWriter Class

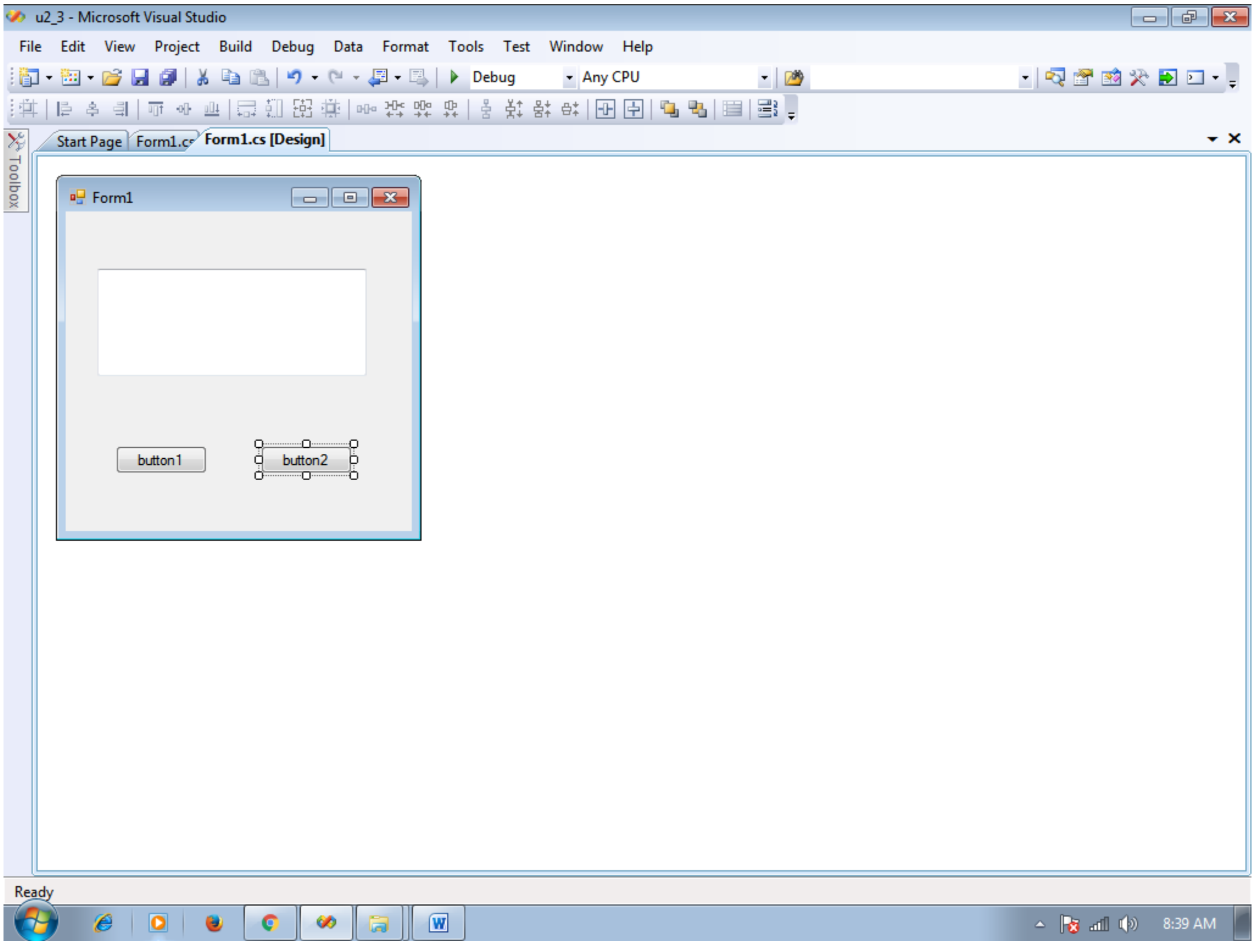
The **StreamWriter** class inherits from the abstract class TextWriter that represents a writer, which can write a series of character.

The following table describes the most commonly used methods of this class –

| Sr.No. | Method & Description |
|---|---|
| 1 | **public override void Close()**<br>Closes the current StreamWriter object and the |

| | |
|---|---|
| | underlying stream. |
| 2 | **public override void Flush()**<br>Clears all buffers for the current writer and causes any buffered data to be written to the underlying stream. |
| 3 | **public virtual void Write(bool value)**<br>Writes the text representation of a Boolean value to the text string or stream. (Inherited from TextWriter.) |
| 4 | **public override void Write(char value)**<br>Writes a character to the stream. |
| 5 | **public virtual void Write(decimal value)**<br>Writes the text representation of a decimal value to the text string or stream. |
| 6 | **public virtual void Write(double value)**<br>Writes the text representation of an 8-byte floating-point value to the text string or stream. |
| 7 | **public virtual void Write(int value)**<br>Writes the text representation of a 4-byte signed integer to the text string or stream. |
| 8 | **public override void Write(string value)**<br>Writes a string to the stream. |
| 9 | **public virtual void WriteLine()**<br>Writes a line terminator to the text string or stream. |

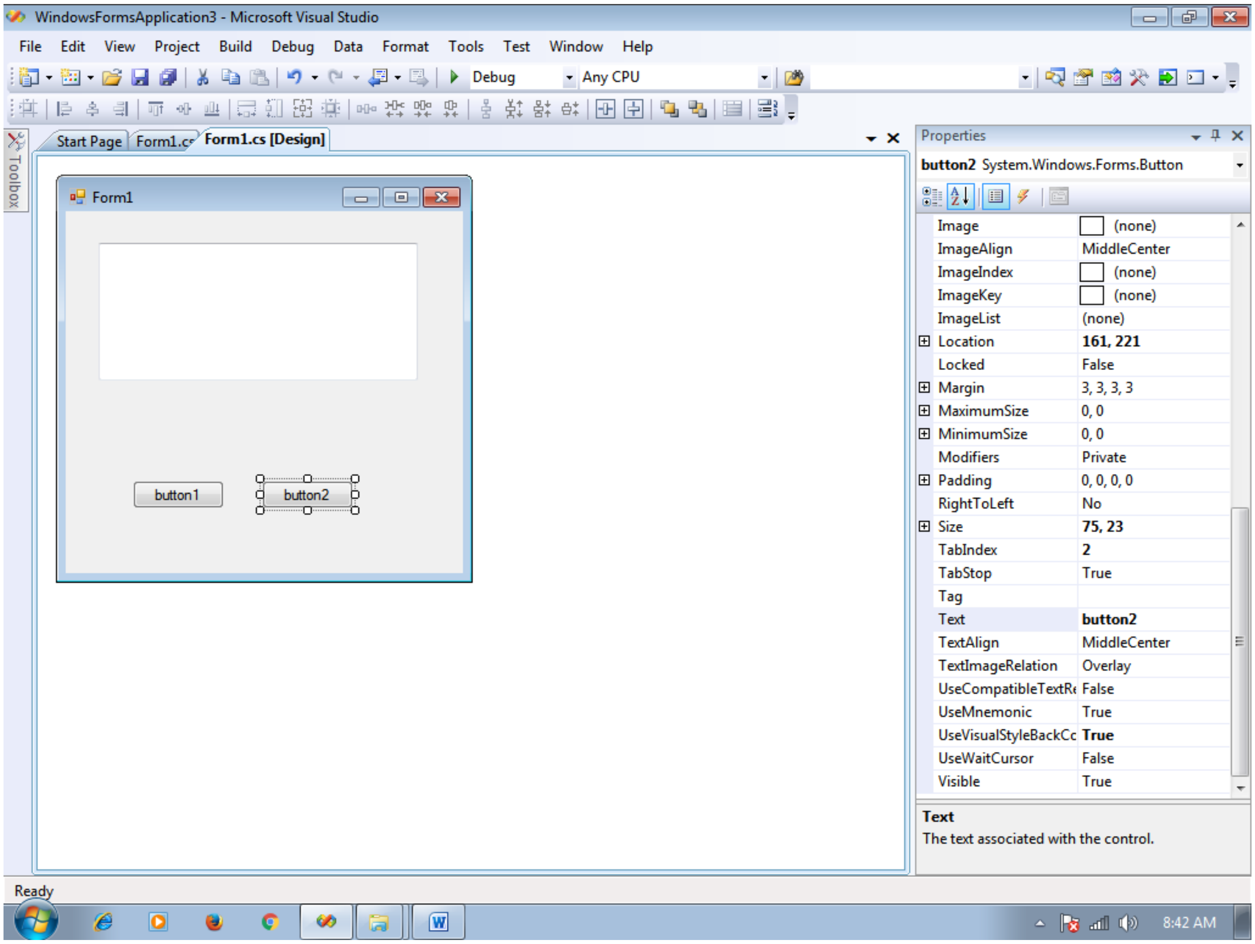
**4. Write a program to read and write text file.**
**(tool box- textbox ,button1 ,button2)**


u2_3 - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Test Window Help
Debug
Any CPU
Toolbox
Start Page
Form1.cs
Form1.cs [Design]
Form1
button1
button2
Ready
8:39 AM

### Code

```
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;
using System.IO;
// import using System.IO;
namespace u2_3
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            StreamWriter sw;
            sw = new StreamWriter("d:\\f1.txt");
            sw.WriteLine("hello");
            sw.WriteLine(textBox1.Text);
            sw.WriteLine("hi hello");
            sw.Close();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            StreamReader sr;
            sr = new StreamReader("d:\\f1.txt");
            textBox1.Text = sr.ReadToEnd();
            sr.Close();

        }
    }
}
```

**5. Write a program to read and write binary file.**


WindowsFormsApplication3 - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Test Window Help
Debug
Any CPU
Toolbox
Start Page
Form1.cs
Form1.cs [Design]
Form1
button1
button2
Properties
Ready
8:42 AM

button2 System.Windows.Forms.Button

| | |
|---|---|
| Image | (none) |
| ImageAlign | MiddleCenter |
| ImageIndex | (none) |
| ImageKey | (none) |
| ImageList | (none) |
| Location | 161, 221 |
| Locked | False |
| Margin | 3, 3, 3, 3 |
| MaximumSize | 0, 0 |
| MinimumSize | 0, 0 |
| Modifiers | Private |
| Padding | 0, 0, 0, 0 |
| RightToLeft | No |
| Size | 75, 23 |
| TabIndex | 2 |
| TabStop | True |
| Tag | |
| Text | button2 |
| TextAlign | MiddleCenter |
| TextImageRelation | Overlay |
| UseCompatibleTextRe | False |
| UseMnemonic | True |
| UseVisualStyleBackCc | True |
| UseWaitCursor | False |
| Visible | True |

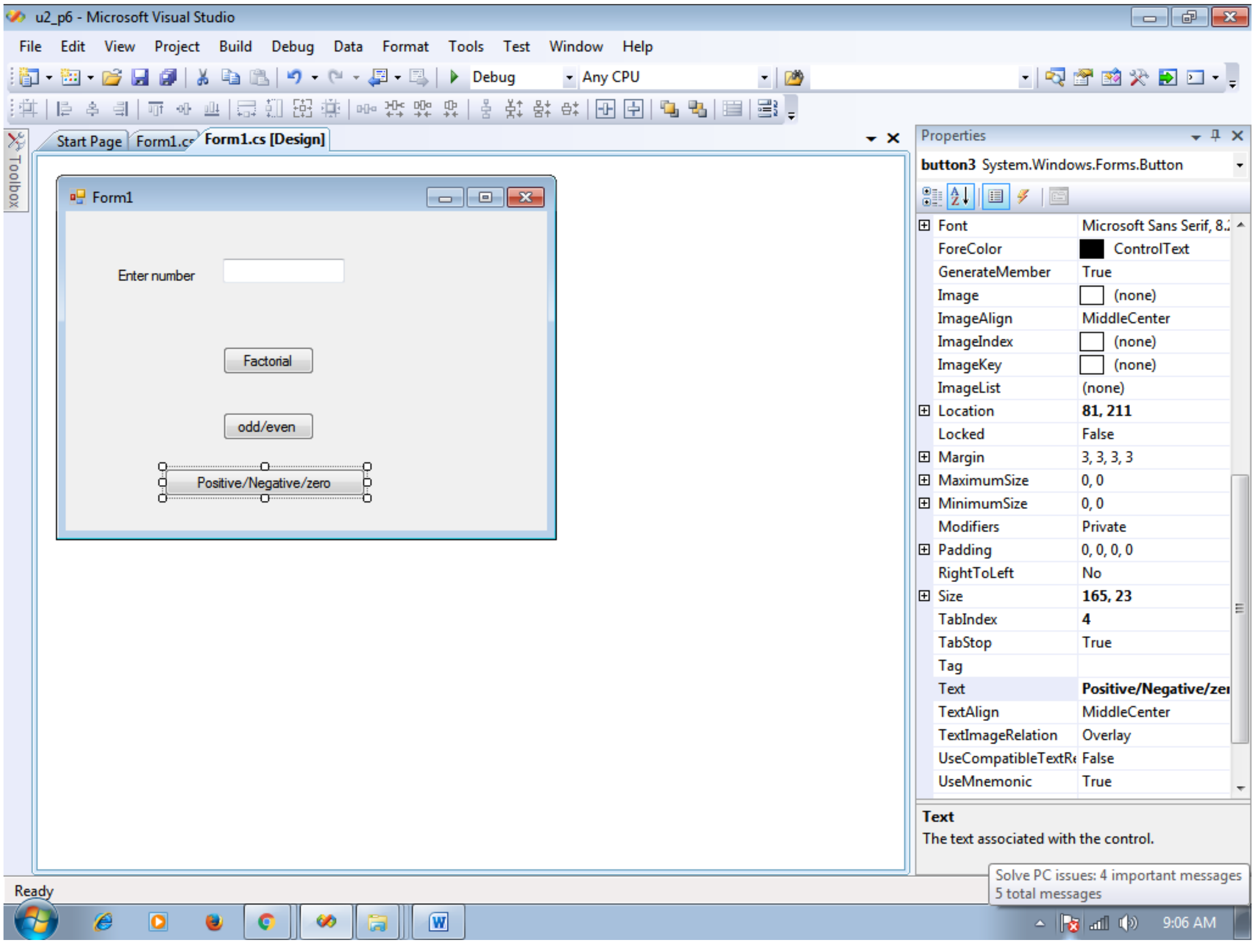
**Text**
The text associated with the control.

## Code

```
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;
using System.IO;

namespace WindowsFormsApplication3
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            FileStream fs = new FileStream("d:\\f2.bin",FileMode.Create);
            BinaryWriter bw;
            bw = new BinaryWriter(fs);

            Int32 i = 12;
            bw.Write(i);
            double d = 34.45;
            bw.Write(d);
            bool b = true;
            bw.Write(b);
            String s = "are";
            bw.Write(s);
            bw.Close();
            fs.Close();
            MessageBox.Show("creates");


        }

        private void button2_Click(object sender, EventArgs e)
        {
            FileStream fs = new FileStream("d:\\f2.bin",FileMode.Open);
            BinaryReader br;
            br = new BinaryReader(fs);
            Int32 i;
            double d;
            bool b;
            String s;
            i = br.ReadInt32();
            textBox1.Text = i.ToString() + Environment.NewLine;

            d = br.ReadDouble();
            textBox1.Text += d.ToString() + Environment.NewLine;

            b= br.ReadBoolean();
            textBox1.Text += b.ToString() + Environment.NewLine;

            s= br.ReadString();

            textBox1.Text += s.ToString() + Environment.NewLine;

            br.Close();
            fs.Close();


        }
    }
}
```

**6. Accept no from user and perform following operation using user defined sub routine functions:**

1) **Factorial of number**
2) **Odd/even**
3) **Positive/ Negative /zero**



button3 System.Windows.Forms.Button

| | |
|---|---|
| Font | Microsoft Sans Serif, 8.2 |
| ForeColor | ControlText |
| GenerateMember | True |
| Image | (none) |
| ImageAlign | MiddleCenter |
| ImageIndex | (none) |
| ImageKey | (none) |
| ImageList | (none) |
| Location | **81, 211** |
| Locked | False |
| Margin | 3, 3, 3, 3 |
| MaximumSize | 0, 0 |
| MinimumSize | 0, 0 |
| Modifiers | Private |
| Padding | 0, 0, 0, 0 |
| RightToLeft | No |
| Size | **165, 23** |
| TabIndex | **4** |
| TabStop | True |
| Tag | |
| Text | **Positive/Negative/zer** |
| TextAlign | MiddleCenter |
| TextImageRelation | Overlay |
| UseCompatibleTextR | False |
| UseMnemonic | True |

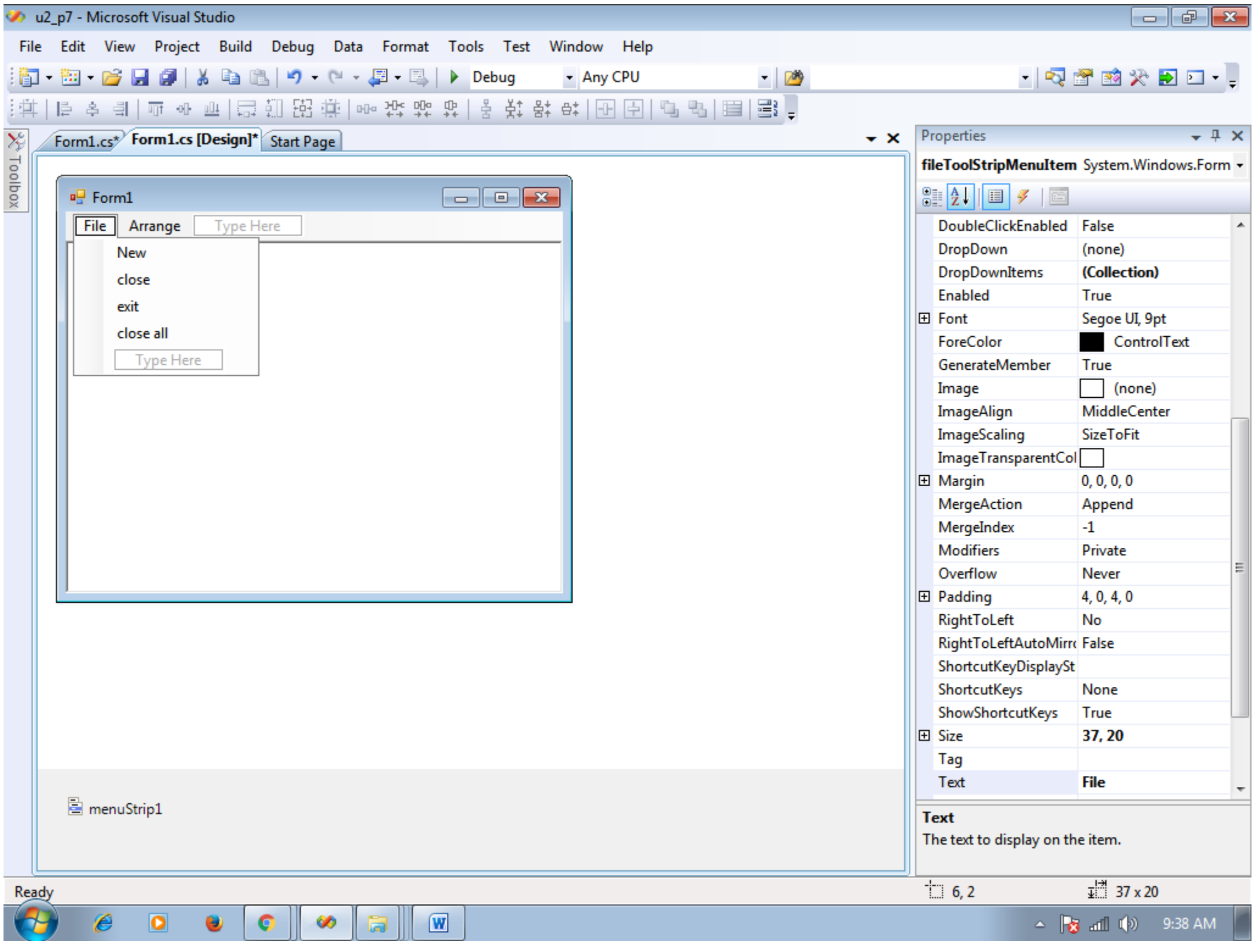
**Text**
The text associated with the control.

Solve PC issues: 4 important messages
5 total messages

## Code

```
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 u2_p6
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Int16 i = Convert.ToInt16(textBox1.Text);
            Int16 r;
            r = fact(i);
            MessageBox.Show(r.ToString());

        }
        public static Int16 fact(Int16 i)
        {
            Int16 n = 1;
            while (i > 0)
            {
                n *=  i;
                i -=  1;
            }
            return n;
        }

        private void button2_Click(object sender, EventArgs e)
        {
            OE(Convert.ToInt16(textBox1.Text));

        }
        public static void OE(Int16 i)
        {
            if (i % 2 == 0)
            {
                MessageBox.Show("Even");
            }
            else
            {
                MessageBox.Show("odd");
            }
        }

        private void button3_Click(object sender, EventArgs e)
        {
            PosNegZero(Convert.ToInt16(textBox1.Text));
        }
        public void PosNegZero(Int16 i)
        {
            if (i < 0)
            {
                MessageBox.Show("negative");
            }
            else if (i == 0)
            {
                MessageBox.Show("zero", "message");

            }
            else
            {
                MessageBox.Show("positive");
            }

        }

    }
}
```

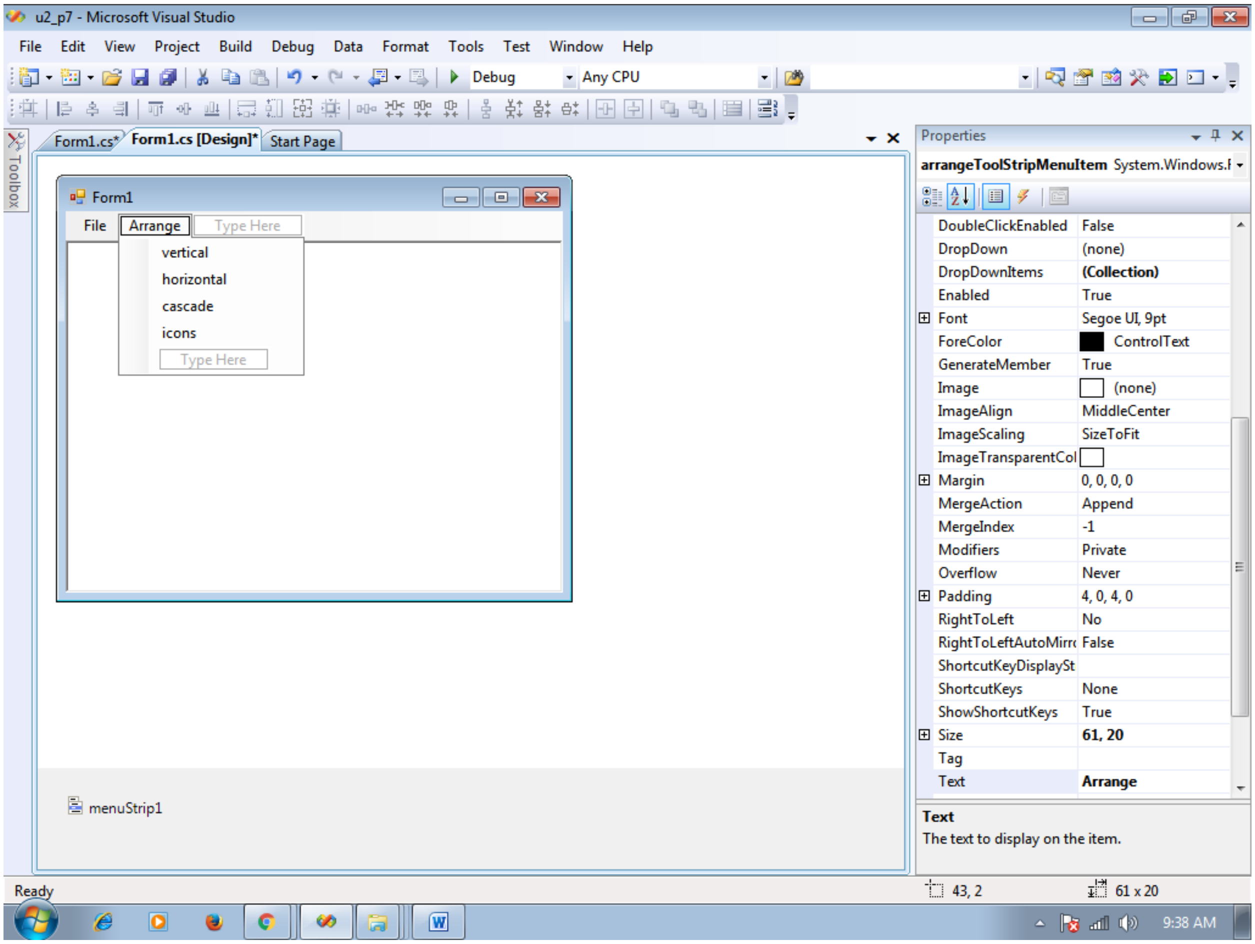
**7. Create MDI form. It must have file menu with option open, close and exit. It should also have window menu to arrange the child forms like tile horizontal, tile vertical, cascade and arrange icons.**



fileToolStripMenuItem System.Windows.Form

| | |
|---|---|
| DoubleClickEnabled | False |
| DropDown | (none) |
| DropDownItems | **(Collection)** |
| Enabled | True |
| ⊞ Font | Segoe UI, 9pt |
| ForeColor | ■ ControlText |
| GenerateMember | True |
| Image | □ (none) |
| ImageAlign | MiddleCenter |
| ImageScaling | SizeToFit |
| ImageTransparentCol | □ |
| ⊞ Margin | 0, 0, 0, 0 |
| MergeAction | Append |
| MergeIndex | -1 |
| Modifiers | Private |
| Overflow | Never |
| ⊞ Padding | 4, 0, 4, 0 |
| RightToLeft | No |
| RightToLeftAutoMirrı | False |
| ShortcutKeyDisplaySt | |
| ShortcutKeys | None |
| ShowShortcutKeys | True |
| ⊞ Size | **37, 20** |
| Tag | |
| Text | **File** |

**Text**
The text to display on the item.



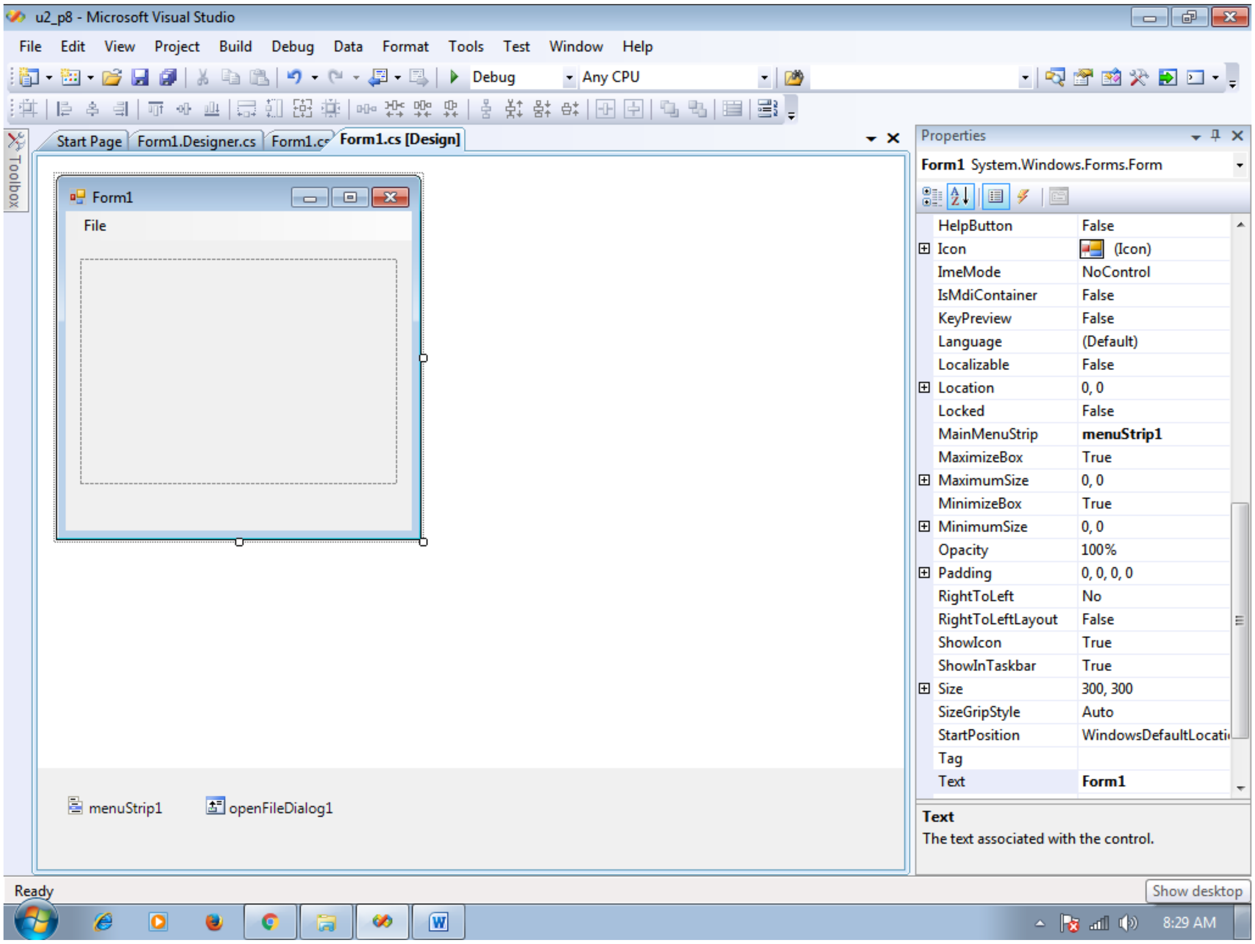
arrangeToolStripMenuItem System.Windows.F

| | |
|---|---|
| DoubleClickEnabled | False |
| DropDown | (none) |
| DropDownItems | **(Collection)** |
| Enabled | True |
| ⊞ Font | Segoe UI, 9pt |
| ForeColor | ■ ControlText |
| GenerateMember | True |
| Image | □ (none) |
| ImageAlign | MiddleCenter |
| ImageScaling | SizeToFit |
| ImageTransparentCol | □ |
| ⊞ Margin | 0, 0, 0, 0 |
| MergeAction | Append |
| MergeIndex | -1 |
| Modifiers | Private |
| Overflow | Never |
| ⊞ Padding | 4, 0, 4, 0 |
| RightToLeft | No |
| RightToLeftAutoMirr | False |
| ShortcutKeyDisplaySt | |
| ShortcutKeys | None |
| ShowShortcutKeys | True |
| ⊞ Size | **61, 20** |
| Tag | |
| Text | **Arrange** |

**Text**
The text to display on the item.

```
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;
using System.IO;

namespace u2_p7
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void newToolStripMenuItem_Click(object sender, EventArgs e)
        {
            Form n;
            n = new Form();
            n.MdiParent = this;
            n.Show();
        }

        private void closeToolStripMenuItem_Click(object sender, EventArgs e)
        {
            //ActivateMdiChild.close();
        }

        private void closeAllToolStripMenuItem_Click(object sender, EventArgs
e)
        {
            foreach (Form f in this.MdiChildren)
            {
                f.Close();
            }
        }

        private void exitToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void verticalToolStripMenuItem_Click(object sender, EventArgs
e)
        {
            this.LayoutMdi(MdiLayout.TileVertical);
        }

        private void horizontalToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            this.LayoutMdi(MdiLayout.TileHorizontal);
        }

        private void cascadeToolStripMenuItem_Click(object sender, EventArgs
e)
        {
            this.LayoutMdi(MdiLayout.Cascade);
        }

        private void iconsToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.LayoutMdi(MdiLayout.ArrangeIcons);
        }
    }
}
```

**8. Create MDI form.it must have file menu with option open, close and exit and one picture box. Allow users to open any picture using open dialog box, that picture should be displayed in picture box.**



| | |
|---|---|
| HelpButton | False |
| Icon | (Icon) |
| ImeMode | NoControl |
| IsMdiContainer | False |
| KeyPreview | False |
| Language | (Default) |
| Localizable | False |
| Location | 0, 0 |
| Locked | False |
| MainMenuStrip | **menuStrip1** |
| MaximizeBox | True |
| MaximumSize | 0, 0 |
| MinimizeBox | True |
| MinimumSize | 0, 0 |
| Opacity | 100% |
| Padding | 0, 0, 0, 0 |
| RightToLeft | No |
| RightToLeftLayout | False |
| ShowIcon | True |
| ShowInTaskbar | True |
| Size | 300, 300 |
| SizeGripStyle | Auto |
| StartPosition | WindowsDefaultLocati |
| Tag | |
| Text | **Form1** |

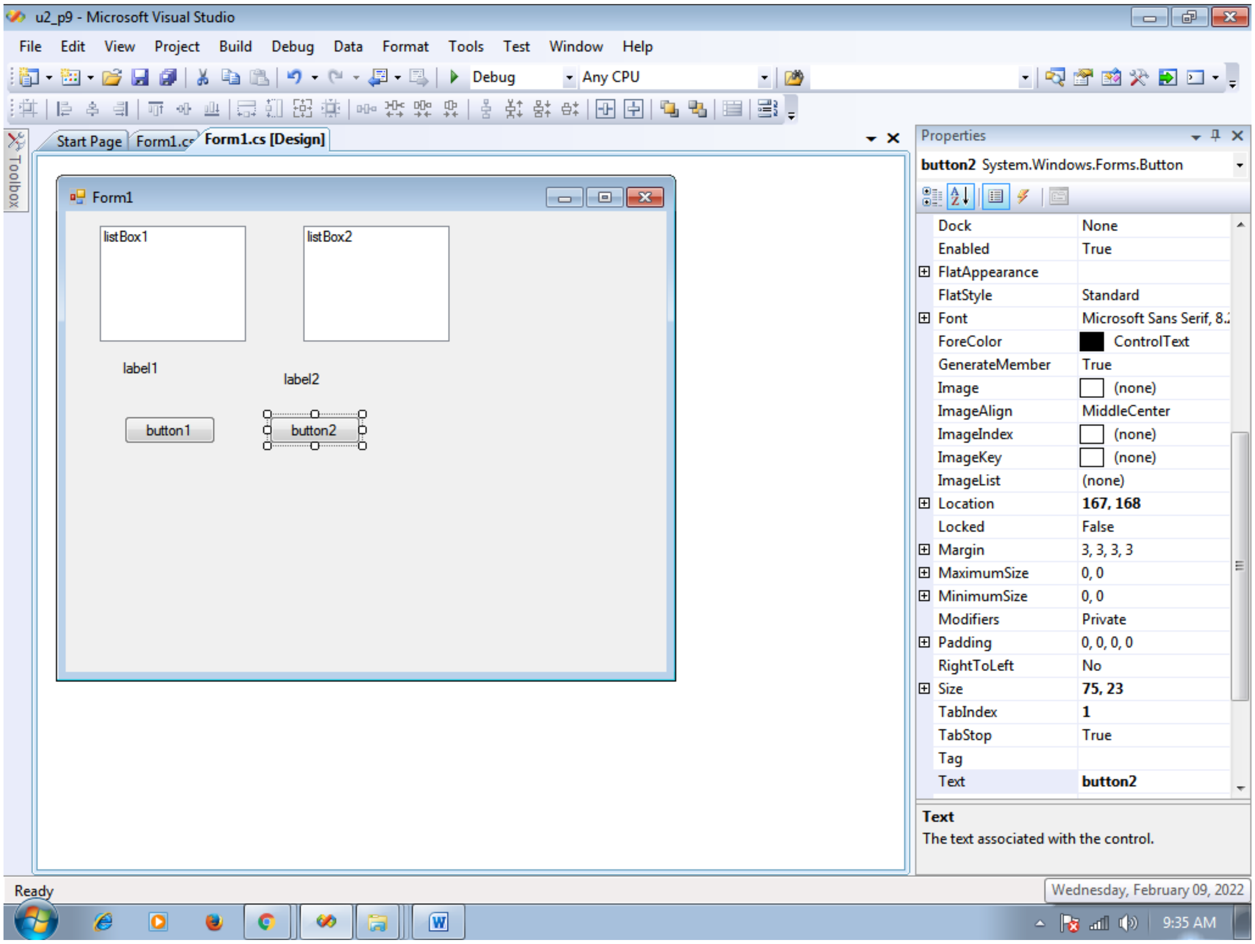
**Text**
The text associated with the control.

## Code

```
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;
using System.IO;

namespace u2_p8
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void openToolStripMenuItem_Click(object sender, EventArgs e)
        {
            openFileDialog1.ShowDialog();
```

```
                pictureBox1.Image = Image.FromFile(openFileDialog1.FileName);
            }

            private void closeToolStripMenuItem_Click(object sender, EventArgs e)
            {
                pictureBox1.Image = null;
            }

            private void exitToolStripMenuItem_Click(object sender, EventArgs e)
            {
                pictureBox1.Image = null;

            }

            private void Form1_Resize(object sender, EventArgs e)
            {
                pictureBox1.Height = this.Height;
                pictureBox1.Width = this.Width;
            }

            private void Form1_Load(object sender, EventArgs e)
            {

            }
        }
    }
```

**9. Write function or subroutine to find maximize, minimize value from an array and also to sort an array.**


u2_p9 - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Test Window Help
Debug
Any CPU
Toolbox
Start Page
Form1.cs
Form1.cs [Design]
Form1
listBox1
listBox2
label1
label2
button1
button2
Properties
Ready
Wednesday, February 09, 2022
9:35 AM

button2 System.Windows.Forms.Button

| | |
|---|---|
| Dock | None |
| Enabled | True |
| ⊞ FlatAppearance | |
| FlatStyle | Standard |
| ⊞ Font | Microsoft Sans Serif, 8. |
| ForeColor | ■ ControlText |
| GenerateMember | True |
| Image | □ (none) |
| ImageAlign | MiddleCenter |
| ImageIndex | □ (none) |
| ImageKey | □ (none) |
| ImageList | (none) |
| ⊞ Location | **167, 168** |
| Locked | False |
| ⊞ Margin | 3, 3, 3, 3 |
| ⊞ MaximumSize | 0, 0 |
| ⊞ MinimumSize | 0, 0 |
| Modifiers | Private |
| ⊞ Padding | 0, 0, 0, 0 |
| RightToLeft | No |
| ⊞ Size | **75, 23** |
| TabIndex | **1** |
| TabStop | True |
| Tag | |
| Text | **button2** |

**Text**
The text associated with the control.

## Code

```
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 u2_p9
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            int[] a = new int[6] { 11, 32, 23, 41, 5, 16 };
            Array.Sort(a);
            for (int i = 0; i <= 4; i++)
            {
                listBox1.Items.Add(a[i].ToString());
            }
            label1.Text = "minimum :" + a[0];
            label2.Text = "maximum :" + a[a.GetUpperBound(0)];
        }

        private void button2_Click(object sender, EventArgs e)
        {
            int[] b = new int[5] { 2, 1, 6, 79, 9 };
            Array.Reverse(b);
            int i;
            for (i = 0; i <= 4; i++)
            {
                listBox2.Items.Add(b[i].ToString());
            }
        }
    }
}
```