**Assignment 2 Unit 2**

Pro 1 : Take 3 Scrollbars including 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 hrlp of scrollbars make 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.

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 U2P1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void hScrollBarRed\_Scroll(object sender, ScrollEventArgs e)

{

pictureBox1.BackColor = Color.FromArgb(hScrollBarRed.Value, hScrollBarGreen.Value, hScrollBarBlue.Value);

label1.Text = Convert.ToString(hScrollBarRed.Value);

}

private void hScrollBarGreen\_Scroll(object sender, ScrollEventArgs e)

{

pictureBox1.BackColor = Color.FromArgb(hScrollBarRed.Value, hScrollBarGreen.Value, hScrollBarBlue.Value);

label2.Text = Convert.ToString(hScrollBarGreen.Value);

}

private void hScrollBarBlue\_Scroll(object sender, ScrollEventArgs e)

{

pictureBox1.BackColor = Color.FromArgb(hScrollBarRed.Value, hScrollBarGreen.Value, hScrollBarBlue.Value);

label3.Text = Convert.ToString(hScrollBarBlue.Value);

}

private void btncolor\_Click(object sender, EventArgs e)

{

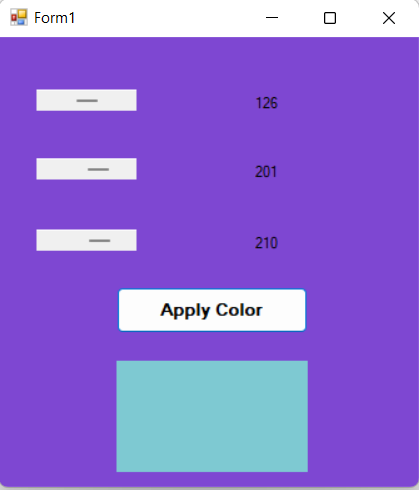
this.BackColor = pictureBox1.BackColor;

}

}

}

Output:



Pro 2: Implement Textpad application using rich textbox. Makes 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.

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 U2P2

{

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.Font, FontStyle.Bold);

}

private void italicToolStripMenuItem\_Click(object sender, EventArgs e)

{

richTextBox1.SelectionFont = new Font(richTextBox1.Font, FontStyle.Italic);

}

private void underlineToolStripMenuItem\_Click(object sender, EventArgs e)

{

richTextBox1.SelectionFont = new Font(richTextBox1.Font, FontStyle.Underline);

}

private void openToolStripMenuItem\_Click(object sender, EventArgs e)

{

OpenFileDialog op = new OpenFileDialog();

//op.tital = "Open";

// junu.filter="Text Document(\*.txt)\\*.txt\All Files(\*.\*)|\*.\*";

if(op.ShowDialog()==DialogResult.OK)

richTextBox1.LoadFile(op.FileName,RichTextBoxStreamType.PlainText);

this.Text=op.FileName;

}

private void saveToolStripMenuItem\_Click(object sender, EventArgs e)

{

SaveFileDialog sf = new SaveFileDialog();

// sf.Tital = "Save";

// junu.filter="Text Document(\*.txt)\\*.txt\All Files(\*.\*)|\*.\*";

if (sf.ShowDialog() == DialogResult.OK)

richTextBox1.SaveFile(sf.FileName, RichTextBoxStreamType.PlainText);

this.Text = sf.FileName;

}

private void exitToolStripMenuItem\_Click(object sender, EventArgs e)

{

Close();

}

private void fontToolStripMenuItem\_Click(object sender, EventArgs e)

{

FontDialog fnt = new FontDialog();

if (fnt.ShowDialog() == DialogResult.OK)

richTextBox1.Font = fnt.Font;

}

private void colorToolStripMenuItem\_Click(object sender, EventArgs e)

{

ColorDialog c = new ColorDialog();

if (c.ShowDialog() == DialogResult.OK)

richTextBox1.ForeColor = c.Color;

}

private void saveAsToolStripMenuItem\_Click(object sender, EventArgs e)

{

SaveFileDialog sf = new SaveFileDialog();

// sf.Tital = "Save";

// junu.filter="Text Document(\*.txt)\\*.txt\All Files(\*.\*)|\*.\*";

if (sf.ShowDialog() == DialogResult.OK)

richTextBox1.SaveFile(sf.FileName, RichTextBoxStreamType.PlainText);

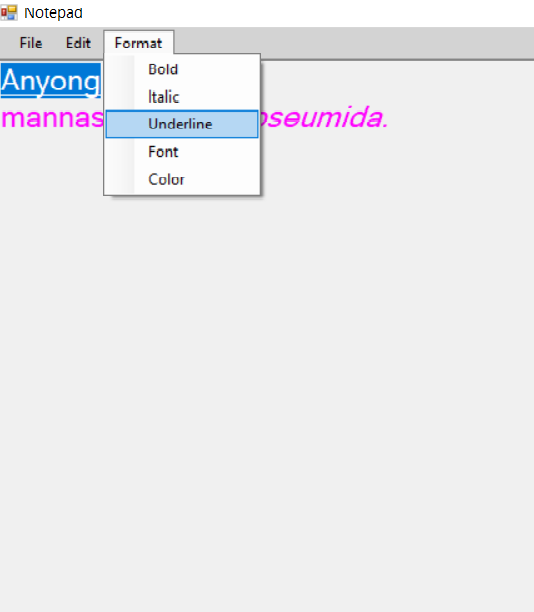
this.Text = sf.FileName;

}

}

}

Output:



Pro 3: Take a Rich Textbox and implement Find, FindNext, Replace, and ReplaceAll functionalities.

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 U2P3

{

public partial class Form1 : Form

{

int pos;

public Form1()

{

InitializeComponent();

}

private void btnfind\_Click(object sender, EventArgs e)

{

pos = Convert.ToInt32(richTextBox1.Find(textBox1.Text, 0, richTextBox1.TextLength, RichTextBoxFinds.None));

if (pos >= 0)

{

richTextBox1.SelectionStart = pos;

richTextBox1.SelectionLength = textBox1.TextLength;

richTextBox1.SelectionColor = Color.Purple;

}

}

private void btnfnext\_Click(object sender, EventArgs e)

{

pos = Convert.ToInt32(richTextBox1.Find(textBox1.Text, pos + 1, richTextBox1.TextLength, RichTextBoxFinds.None));

if (pos >= 0)

{

richTextBox1.SelectionStart = pos;

richTextBox1.SelectionLength = textBox1.TextLength;

richTextBox1.SelectionColor = Color.Purple;

}

}

private void btnrep\_Click(object sender, EventArgs e)

{

richTextBox1.SelectedText = textBox2.Text;

}

private void btnrall\_Click(object sender, EventArgs e)

{

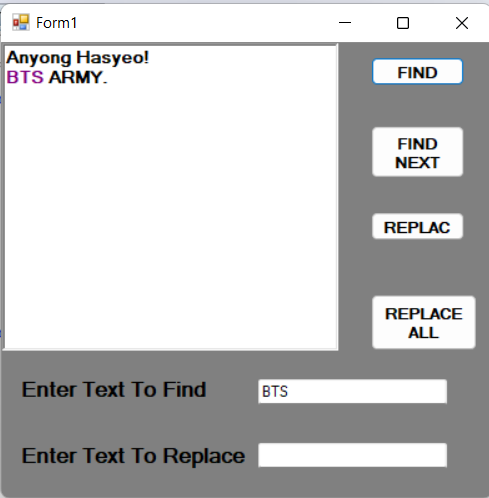
richTextBox1.Text = richTextBox1.Text.Replace(textBox1.Text, textBox2.Text);

}

}

}

Output:



Pro 4: Write a Program to Read and Write Text File.

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 U2P4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

FileStream fc = new FileStream(@"E:\C#\Unit2\Demo.txt", FileMode.Create);

StreamWriter sw = new StreamWriter(fc);

sw.Write(DateTime.Now + "\r\n" + textBox1.Text);

sw.Close();

fc.Close();

}

private void button2\_Click(object sender, EventArgs e)

{

FileStream fc = new FileStream(@"E:\C#\Unit2\Demo.txt", FileMode.Open);

StreamReader sr = new StreamReader(fc);

textBox2.Text = textBox2.Text + "Content of file is as follow:" + sr.ReadToEnd();

sr.Close();

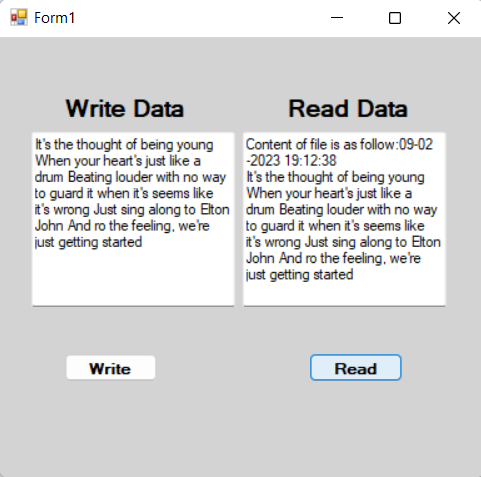
fc.Close();

}

}

}

Output:



Pro 5: Write a program to Read and Write Binary File.

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 U2P5

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void btnwrite\_Click(object sender, EventArgs e)

{

FileStream fc = new FileStream(@"E:\C#\Unit2\Demo2.dat", FileMode.Create);

BinaryWriter bw = new BinaryWriter(fc);

bw.Write(Convert.ToInt32(textBox1.Text));

bw.Write(textBox2.Text);

bw.Write(Convert.ToDouble(textBox3.Text));

bw.Close();

fc.Close();

MessageBox.Show("Data written to binary file Successfully", "Informtion", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

private void btnread\_Click(object sender, EventArgs e)

{

FileStream fc = new FileStream(@"E:\C#\Unit2\Demo2.dat", FileMode.Open);

BinaryReader br = new BinaryReader(fc);

int s = br.ReadInt32();

String s1 = br.ReadString();

Double cont = br.ReadDouble();

labOutput.Text = labOutput.Text +"\n"+ "Student`s RollNo:" + s+"\n"+ "Student`s Name:" + s1+"\n"+ "Student`s ContactNo:" + cont;

br.Close();

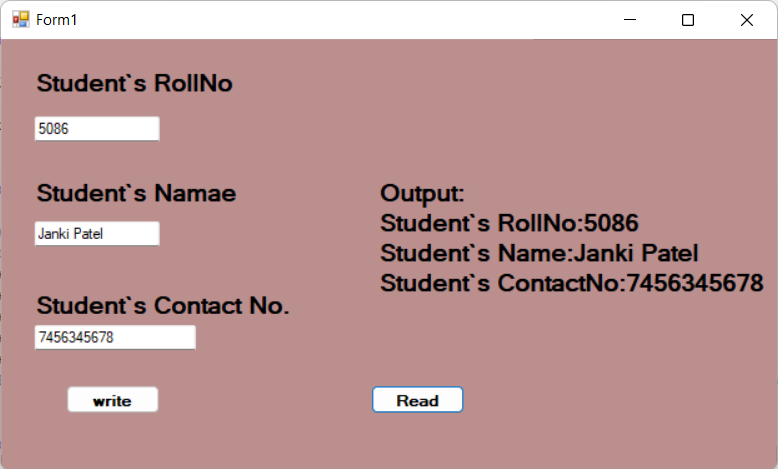
fc.Close();

}

}

}

Output:



Pro 6: Accept no form user and perform following operations using defined sub routines or functions: 1. Factorial of number 2. Odd/Even

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 U2P6

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

public int fact()

{

int f = 1;

for (int i = 1; i <= int.Parse(textBox1.Text); i++)

{

f = f \* i;

}

return f;

}

private void btnFact\_Click(object sender, EventArgs e)

{

labFactOut.Text = fact().ToString();

}

public int oddEven()

{

int i=int.Parse(textBox1.Text);

if (i % 2 == 0)

{

MessageBox.Show("Number is Even", "Infrmation", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("Number is Odd", "Infrmation", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

return i;

}

private void btnOddEven\_Click(object sender, EventArgs e)

{

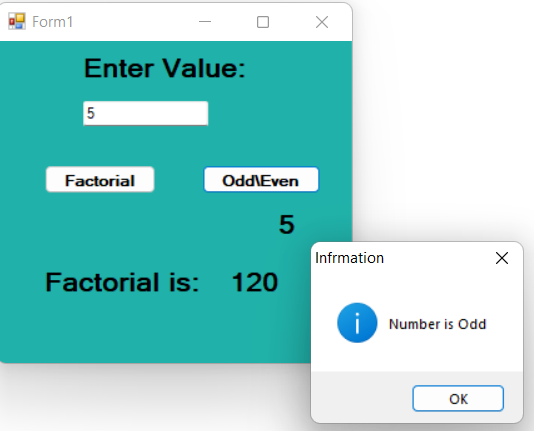
labOddEven.Text = oddEven().ToString();

}

}

}

Output:



Pro 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.

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 U2P7

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void openToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form2 child1 = new Form2();

child1.MdiParent = this;

child1.Show();

Form3 child2 = new Form3();

child2.MdiParent = this;

child2.Show();

Form4 child3 = new Form4();

child3.MdiParent = this;

child3.Show();

}

private void cascadeToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.LayoutMdi(MdiLayout.Cascade);

}

private void tileHorizontalToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.LayoutMdi(MdiLayout.TileHorizontal);

}

private void tileVerticalToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.LayoutMdi(MdiLayout.TileVertical);

}

private void arrangeIconsToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.LayoutMdi(MdiLayout.ArrangeIcons);

}

private void closeToolStripMenuItem\_Click(object sender, EventArgs e)

{

foreach (Form f1 in this.MdiChildren)

{

f1.Close();

}

}

private void exitToolStripMenuItem\_Click(object sender, EventArgs e)

{

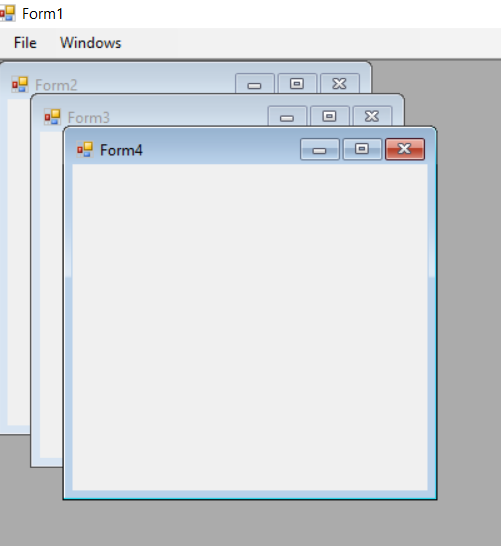
Application.Exit();

}

}

}

Output:



Pro 8: Create MDI form. It must have file menu with option Open, Close, and Exit and one picture box. Allow user to open any picture using open dialog box, that picture should be displayed in the picture box.

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 U2P8

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void openToolStripMenuItem\_Click(object sender, EventArgs e)

{

DialogResult result = openFileDialog1.ShowDialog();

if (result == DialogResult.OK)

{

openFileDialog1.Filter = "jpg files(\*.jpg) | \*.jpg";

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

}

}

private void closeToolStripMenuItem\_Click(object sender, EventArgs e)

{

Close();

}

private void exitToolStripMenuItem\_Click(object sender, EventArgs e)

{

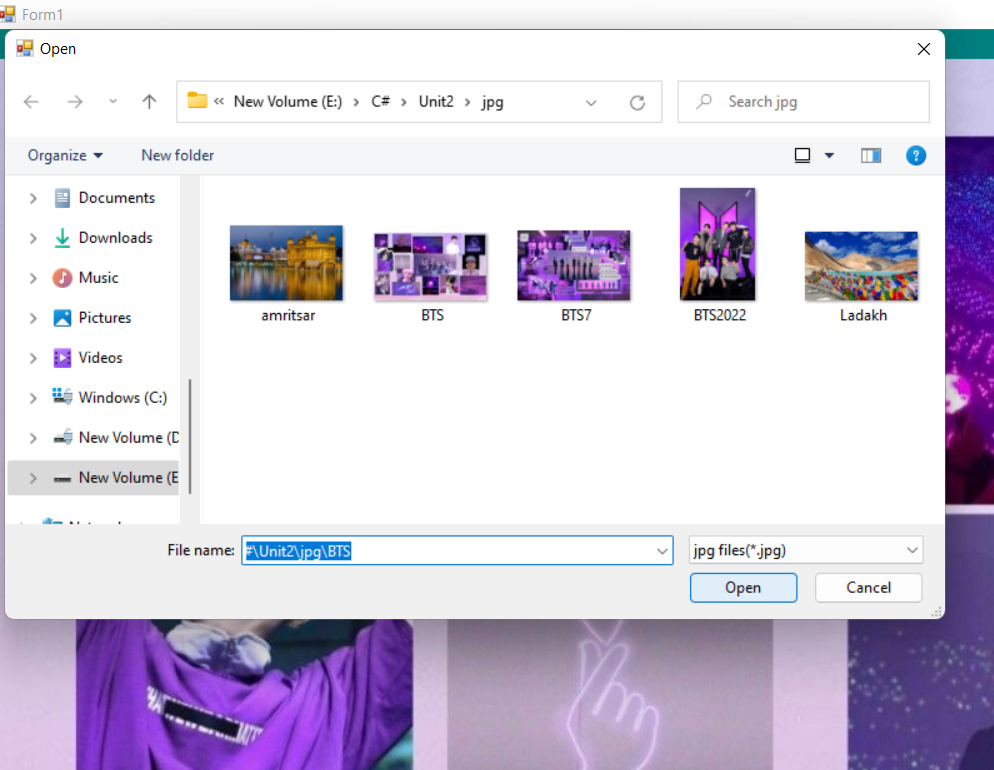
Application.Exit();

}

}

}

Output:



Pro 9: Write function or subroutine to find maximum, minimum value from an array and also to sort an array.

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 U2P9

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

int[] n = new int[5];

private void btndisplay\_Click(object sender, EventArgs e)

{

n[0] = 10;

n[1] = 5;

n[2] = 7;

n[3] = 31;

n[4] = 30;

for (int i = 0; i < 5; i++)

textBox1.Text += Convert.ToString(n[i]) + "\t\n";

}

private void btnSort\_Click(object sender, EventArgs e)

{

Array.Sort(n);

for (int i = 0; i < 5; i++)

textBox2.Text += Convert.ToString(n[i]) + "\t\n";

}

public int min()

{

n[0] = 10;

n[1] = 5;

n[2] = 7;

n[3] = 31;

n[4] = 30;

int min = n[0];

for (int i = 0; i < 5; i++)

{

if (n[i] < min)

min = n[i];

}

return min;

}

private void btnmin\_Click(object sender, EventArgs e)

{

labMinMax.Text = "Minimum Value:" + min().ToString();

}

public int max()

{

n[0] = 10;

n[1] = 5;

n[2] = 7;

n[3] = 31;

n[4] = 30;

int max = n[0];

for (int i = 0; i < 5; i++)

{

if (n[i] > max)

max = n[i];

}

return max;

}

private void btnMax\_Click(object sender, EventArgs e)

{

labMinMax.Text = "Maximum Value:" + max().ToString();

}

}

}

Output:

