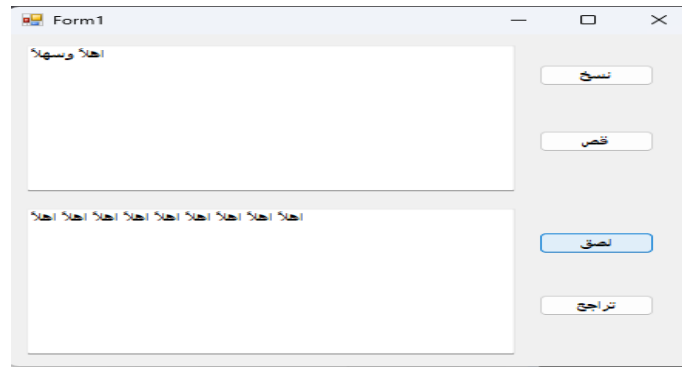


بسم الله الرحمن الرحيم

الاسم : محمد نشوان الفقيه

البرنامج الاول

تنفيذ امر اللصق والنسخ



تنفيذ امر التراجع عن كل التغييرات ابتداءً من اخر تغيير



الكود

```
using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace lect8
{
    public partial class Form1 : Form
```

```

{

private Stack<TextState> undoStack = new Stack<TextState>();
private Stack<TextState> redoStack = new Stack<TextState>();
private bool isUndoRedoOperation = false;

public Form1()
{
    InitializeComponent();
}

private void Form1_Load(object sender, EventArgs e)
{
    txtSource.Text = "اهلاً وسهلاً";
    txtDestination.Text = "يمكنك اللصق هنا";
    txtSource.TextChanged += new EventHandler(TextChangedHandler);
    txtDestination.TextChanged += new EventHandler(TextChangedHandler);
}

private class TextState
{
    public string SourceText
    {
        get; set;
    }
    public string DestinationText
    {
        get; set;
    }

    public TextState(string sourceText, string destinationText)
    {
        SourceText = sourceText;
        DestinationText = destinationText;
    }
}

private void SaveState()
{
    if(!isUndoRedoOperation)
    {
        undoStack.Push(new TextState(txtSource.Text, txtDestination.Text));
        redoStack.Clear();
    }
}

private void TextChangedHandler(object sender, EventArgs e)
{
    SaveState();
}

private void btnCut_Click(object sender, EventArgs e)
{
    if(!string.IsNullOrEmpty(txtSource.SelectedText))
    {
        SaveState();
        Clipboard.SetText(txtSource.SelectedText);
    }
}
}

```

```

        txtSource.SelectedText = "";
    }
}

private void btnCopy_Click(object sender, EventArgs e)
{
    if(!string.IsNullOrEmpty(txtSource.SelectedText))
    {
        Clipboard.SetText(txtSource.SelectedText);
    }
}

private void btnPaste_Click(object sender, EventArgs e)
{
    if(Clipboard.ContainsText())
    {
        SaveState();
        txtDestination.Text += Clipboard.GetText();
    }
}

private void btnUndo_Click(object sender, EventArgs e)
{
    if(undoStack.Count > 1)
    {
        isUndoRedoOperation = true;
        redoStack.Push(undoStack.Pop());
        TextState previousState = undoStack.Peek();

        txtSource.Text = previousState.SourceText;
        txtDestination.Text = previousState.DestinationText;

        isUndoRedoOperation = false;
    }
}

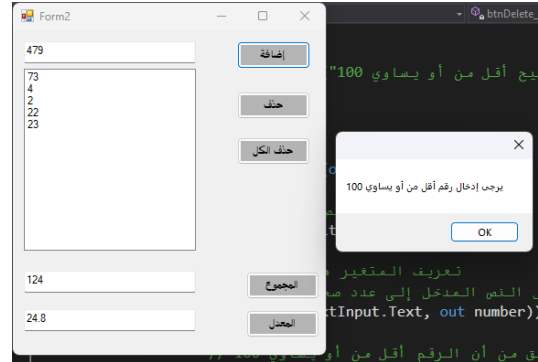
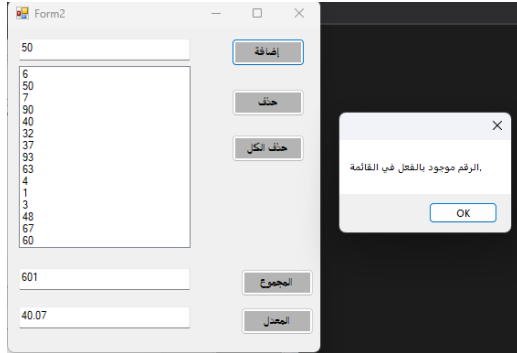
private void btnRedo_Click(object sender, EventArgs e)
{
    if(redoStack.Count > 0)
    {
        isUndoRedoOperation = true;
        undoStack.Push(redoStack.Pop());
        TextState nextState = redoStack.Peek();

        txtSource.Text = nextState.SourceText;
        txtDestination.Text = nextState.DestinationText;

        isUndoRedoOperation = false;
    }
}
}
}
}

```

البرنامج الثاني



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace lect8
{
    public partial class Form2 : Form
    {
        List<int> numbers = new List<int>();

        public Form2()
        {
            InitializeComponent();
            txtInput.KeyPress += TxtInput_KeyPress;
            txtInput.TextChanged += TxtInput_TextChanged;
        }

        private void TxtInput_KeyPress(object sender, KeyPressEventArgs e)
        {
            if(!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar))
            {
                e.Handled = true;
            }
        }

        private void TxtInput_TextChanged(object sender, EventArgs e)
        {
            btnAdd.Enabled = !string.IsNullOrEmpty(txtInput.Text);
        }

        private void Form2_Load(object sender, EventArgs e)
        {
            Random random = new Random();
            while(numbers.Count < 10)
            {
                int randomNumber = random.Next(1, 101);
```

```

        if(!numbers.Contains(randomNumber))
        {
            numbers.Add(randomNumber);
        }
    }
    UpdateList();

    btnAdd.Enabled = false;
}

private void btnAdd_Click(object sender, EventArgs e)
{
    int number = int.Parse(txtInput.Text.Trim());
    if(number <= 100)
    {
        if(!numbers.Contains(number))
        {
            numbers.Add(number);
            if(numbers.Count > 20)
            {
                numbers.RemoveAt(0);
            }
            UpdateList();
        }
        else
        {
            MessageBox.Show("الرقم موجود بالفعل في القائمة");
        }
    }
    else
    {
        MessageBox.Show("يرجى إدخال رقم أقل من أو يساوي 100");
    }

    txtInput.Clear();
}

private void btnDelete_Click(object sender, EventArgs e)
{
    if(lstNumbers.SelectedItem != null)
    {
        int selectedNumber = int.Parse(lstNumbers.SelectedItem.ToString());
        numbers.Remove(selectedNumber);
        UpdateList();
    }
}

private void btnClear_Click(object sender, EventArgs e)
{
    numbers.Clear();
    UpdateList();
}

private void btnSum_Click(object sender, EventArgs e)
{
    int sum = numbers.Sum();
    txtSum.Text = sum.ToString();
}

```

```
private void btnAverage_Click(object sender, EventArgs e)
{
    if(numbers.Count > 0)
    {
        double average = numbers.Average();
        txtAverage.Text = average.ToString("0.##");
    }
}

private void UpdateList()
{
    lstNumbers.Items.Clear();
    foreach(var number in numbers.Take(20))
    {
        lstNumbers.Items.Add(number);
    }
}
}
```

