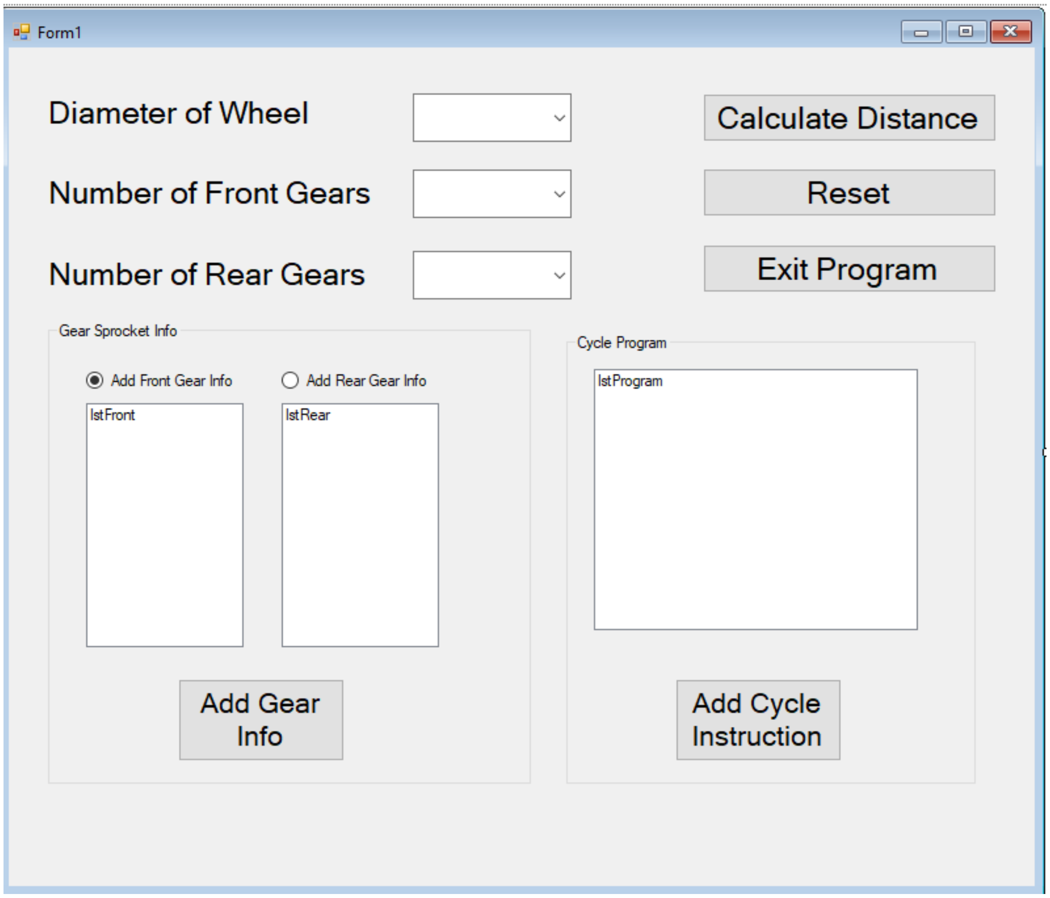
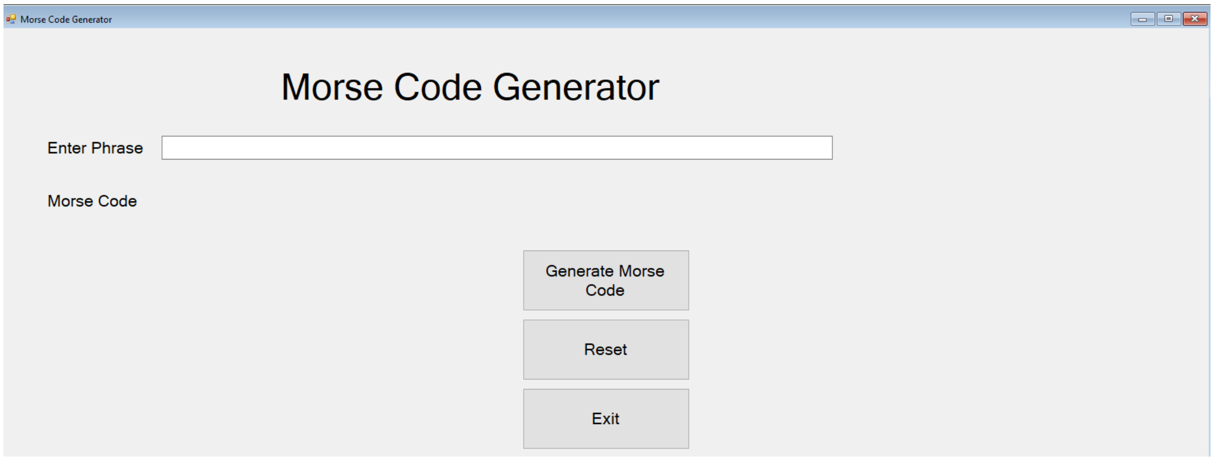


```
//SpinCycle Windows Application
//Author: nmessa
//Date: 6/6/2022

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 Microsoft.VisualBasic;
```

```
namespace SpinCycle
{
   public partial class Form1 : Form
   {
      //Global variables
      int crankSprockets, wheelSprockets;
      double diameter;
      int rotations, gear;
      double distance = 0;

      public Form1()
      {
         InitializeComponent();
      }

      private void cboRG_SelectedIndexChanged(object sender, EventArgs e)
      {
         //Get the number number of sprockets on the wheel
         wheelSprockets = Convert.ToInt32(cboRG.Text);

         //If all combo boxes populated enable the Add Gear Info Button
         if (cboDiameter.Text != "" && cboFG.Text != "" && cboRG.Text != "")
            btnAdd.Enabled = true;
      }

      private void btnAdd_Click(object sender, EventArgs e)
      {
         string temp;
         if (radFront.Checked)   //Add gears (Tooth count) on crank
         {
            for (int i = 0; i < crankSprockets; i++)
            {
               temp = Interaction.InputBox("Enter number of gear teeth", "Gear Teeth",
               "", 10, 10);
               lstFront.Items.Add(temp);
            }
         }

         if (radRear.Checked)  //Add gears (Tooth count) on wheel
         {
            for (int i = 0; i < wheelSprockets; i++)
            {
               temp = Interaction.InputBox("Enter number of gear teeth", "Gear Teeth",
               "", 10, 10);
               lstRear.Items.Add(temp);
            }
         }
         //If lstFront and lstRear populated disable the Add Gear Info button
         //and enable the Add cycle instruction button
         if (lstFront.Items.Count == crankSprockets &&
            lstRear.Items.Count == wheelSprockets)
         {
            btnAdd.Enabled = false;
            btnProgram.Enabled = true;
         }
      }

      private void cboDiameter_SelectedIndexChanged(object sender, EventArgs e)
      {
         //Get wheel diameter from combobox
         diameter = Convert.ToDouble(cboDiameter.Text);
      }

      private void cboFG_SelectedIndexChanged(object sender, EventArgs e)
      {
         //Get the numfer of sprockets on crank from combobox
         crankSprockets = Convert.ToInt32(cboFG.Text);
      }

      private void btnProgram_Click(object sender, EventArgs e)
      {
         //Get cycle instructions and place in lstProgram
         string temp;
         temp = Interaction.InputBox("Enter the gear number and number of turns",
            "Cycle Program", "", 10, 10);
         lstProgram.Items.Add(temp);
         btnCalculate.Enabled = true;
      }

      private void btnExit_Click(object sender, EventArgs e)
      {
         MessageBox.Show("Thank you for using SpinCycle");
         this.Close();
      }
```

```
private void btnReset_Click(object sender, EventArgs e)
{
    //Reset the program
    cboDiameter.Text = "";
    cboFG.Text = "";
    cboRG.Text = "";
    lstFront.Items.Clear();
    lstRear.Items.Clear();
    lstProgram.Items.Clear();
    radFront.Checked = true;
    radRear.Checked = false;
    btnAdd.Enabled = false;
    btnCalculate.Enabled = false;
    btnProgram.Enabled = false;

}

private void btnCalculate_Click(object sender, EventArgs e)
{
    //Create three Lsts for store crank and wheel gear data as gear ratios
    List<double> ratios = new List<double>();
    List<int> crankGears = new List<int>();
    List<int> wheelGears = new List<int>();

    //Store crankGears
    for (int i = 0; i < crankSprockets; i++)
    {
        lstFront.SelectedIndex = i;
        crankGears.Add(Convert.ToInt32(lstFront.SelectedItem));
    }

    //Store wheelGears
    for (int i = 0; i < wheelSprockets; i++)
    {
        lstRear.SelectedIndex = i;
        wheelGears.Add(Convert.ToInt32(lstRear.SelectedItem));
    }

    //Calculate gear ratios
    for (int i = 0; i < crankGears.Count; i++)
    {
        for (int j = 0; j < wheelGears.Count; j++)
        {
            ratios.Add(1.0 * crankGears[i] / wheelGears[j]);
        }
    }
```

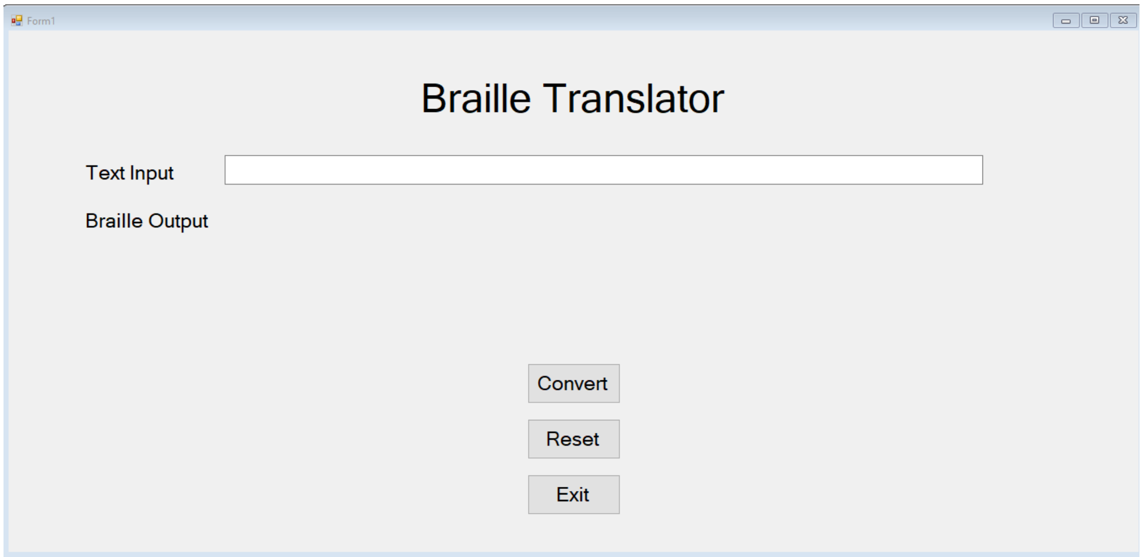
```
            //Sorts gear ratios
            ratios.Sort();

            //Print Gear Table
            string newline = Environment.NewLine;
            string message = "Gear" + "\t" + "Ratio" + newline;
            for (int i = 0; i <  crankSprockets * wheelSprockets; i++)
            {
                message += (i+1) + "\t" + Math.Round(ratios[i], 5) + newline;
            }
            MessageBox.Show(message);

            //Read in gear and rotation data
            string[] temp;
            for (int i = 0; i < lstProgram.Items.Count; i++)
            {
                lstProgram.SelectedIndex = i;
                temp = lstProgram.SelectedItem.ToString().Split(' ');
                gear = Convert.ToInt32(temp[0]);
                rotations = Convert.ToInt32(temp[1]);

                //Calculate distance traveled
                distance += diameter * ratios[gear - 1] * rotations * Math.PI;
            }

            //Convert inches to feet
            distance /= 12.0;

            //Output result to a label
            lblOutput.Text = "Distance traveled: " + Math.Round(distance, 2) + " feet";
        }
    }
}
```



//Morse Code Generator Windows Application
//Author: nmessa
//Date: 6/6/2022

```
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 MorseCode
{
   public partial class Form1 : Form
   {
      //Create an array with all of the Morse codes for the letters A to Z
      string[] codes = {".-", "-...", "-.-.", "-..", ".", "..-.", "--.","....", "..", ".---",
                        "-.-", ".-..", "--", "-.", "---", ".--.", "--.-", ".-.", "...",
                        "-", "..-", "...-", ".--", "-..-", "-.--", "--.."};
      string phrase;
      string morse = "";

      public Form1()
      {
         InitializeComponent();
      }
```

```
        private void btnGenerate_Click(object sender, EventArgs e)
        {
            phrase = txtInput.Text.ToUpper();
            for (int i = 0; i < phrase.Length; i++)
            {
                if (Char.IsLetter(phrase[i]))  //Encode all of the letters
                {
                    morse += codes[phrase[i] - 65];
                }
                else //encode punctuations
                {
                    switch (phrase[i])
                    {
                        case '_':
                            morse += "..--";
                            break;
                        case '.':
                            morse += "---.";
                            break;
                        case ',':
                            morse += ".-.-";
                            break;
                        case '?':
                            morse += "----";
                            break;
                        case ' ':
                            morse += " ";
                            break;
                    }
                }
                morse += " ";
            }
            lblMorse.Text = morse;
        }

        private void btnReset_Click(object sender, EventArgs e)
        {
            lblMorse.Text = "";
            txtInput.Text = "";
            txtInput.Focus();
        }

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

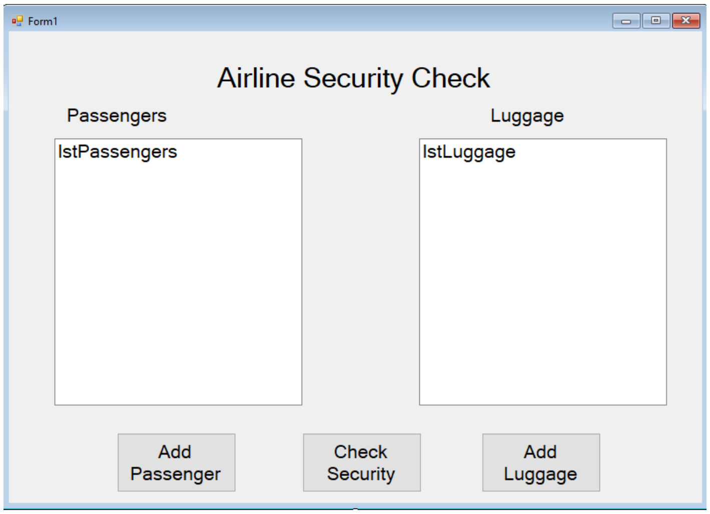


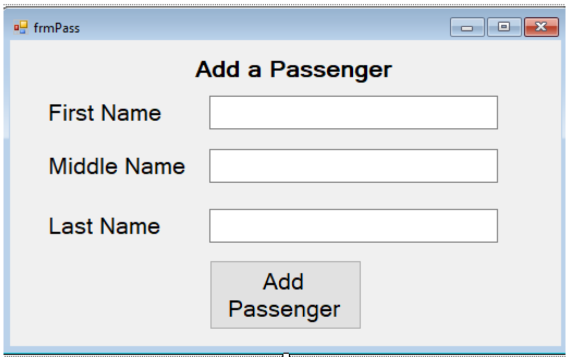
```
//Braille Printer Windows Application
//Author: nmessa
//Date: 6/6/2022

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 BraillePrint
{
    public partial class Form1 : Form
    {
        List<string> braille = new List<string>();

        public Form1()
        {
            InitializeComponent();
        }
```

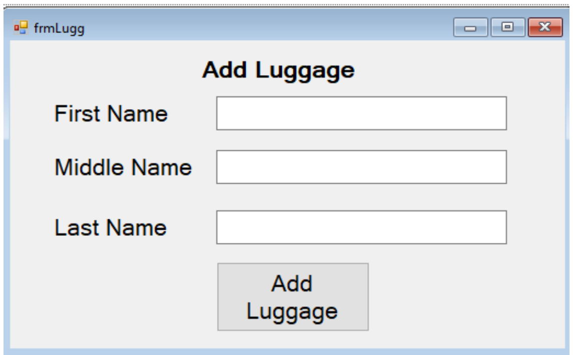
```
private void btnConvert_Click(object sender, EventArgs e)
{
   //Define Braille characters
   //Note the " character must be defined as \"
   string b = " a1b'k2l@cif/msp\"e3h9o6r^djg>ntq'*5<-u8v.%[$+x!&;:4\0z7(_?w]#y)=";
   string phrase;
   string braillePhrase = "";
   int index;

   phrase = txtInput.Text.ToLower(); ;

   //Generate the Braille Phrase
   for (int i = 0; i < phrase.Length; i++)
   {
      index = b.IndexOf(phrase[i]);
      braillePhrase += (braille[index] + " ");
   }

   //Display the Braille phrase
   lblOutput.Text = braillePhrase;
}

private void Form1_Load(object sender, EventArgs e)
{
   //Store Braille Character set
   braille.Add("\u2800");
   braille.Add("\u2801");
   braille.Add("\u2802");
   braille.Add("\u2803");
   braille.Add("\u2804");
   braille.Add("\u2805");
   braille.Add("\u2806");
   braille.Add("\u2807");
   braille.Add("\u2808");
   braille.Add("\u2809");
   braille.Add("\u280A");
   braille.Add("\u280B");
   braille.Add("\u280C");
   braille.Add("\u280D");
   braille.Add("\u280E");
   braille.Add("\u280F");
   braille.Add("\u2810");
   braille.Add("\u2811");
   braille.Add("\u2812");
   braille.Add("\u2813");
   braille.Add("\u2814");
   braille.Add("\u2815");
   braille.Add("\u2816");
```

```
    braille.Add("\u2817");
    braille.Add("\u2818");
    braille.Add("\u2819");
    braille.Add("\u281A");
    braille.Add("\u281B");
    braille.Add("\u281C");
    braille.Add("\u281D");
    braille.Add("\u281E");
    braille.Add("\u281F");
    braille.Add("\u2820");
    braille.Add("\u2821");
    braille.Add("\u2822");
    braille.Add("\u2823");
    braille.Add("\u2824");
    braille.Add("\u2825");
    braille.Add("\u2826");
    braille.Add("\u2827");
    braille.Add("\u2828");
    braille.Add("\u2829");
    braille.Add("\u282A");
    braille.Add("\u282B");
    braille.Add("\u282C");
    braille.Add("\u282D");
    braille.Add("\u282E");
    braille.Add("\u282F");
    braille.Add("\u2830");
    braille.Add("\u2831");
    braille.Add("\u2832");
    braille.Add("\u2833");
    braille.Add("\u2834");
    braille.Add("\u2835");
    braille.Add("\u2836");
    braille.Add("\u2837");
    braille.Add("\u2838");
    braille.Add("\u2839");
    braille.Add("\u283A");
    braille.Add("\u283B");
    braille.Add("\u283C");
    braille.Add("\u283D");
    braille.Add("\u283E");
    braille.Add("\u283F");
}
```

```
        private void btnReset_Click(object sender, EventArgs e)
        {
            lblOutput.Text = "";
            txtInput.Text = "";
            txtInput.Focus();
        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            MessageBox.Show("Thank you for usint the Braille translator");
            this.Close();
        }
    }
}
```







```
//Airline Security System
//Author: nmessa
//Date: 6.6.2022

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 Security
{
    public partial class frmMain : Form
    {
        public static string first;
        public static string middle;
        public static string last;

        public frmMain()
        {
            InitializeComponent();
        }

        private void btnAddPass_Click(object sender, EventArgs e)
        {
            FileStream fs = new FileStream("passengers.txt", FileMode.Append, FileAccess.Write);
            StreamWriter writer = new StreamWriter(fs);
            string name;
            frmPass pass = new frmPass();
            pass.ShowDialog();
            name = first + " " + middle + " " + last;
            lstPassengers.Items.Add(name);
            writer.WriteLine(name);
            writer.Close();
            fs.Close();
        }

        private void btnAddLuggage_Click(object sender, EventArgs e)
        {
            FileStream fs = new FileStream("luggage.txt", FileMode.Append, FileAccess.Write);
            StreamWriter writer = new StreamWriter(fs);
            string name;
            frmLugg lugg = new frmLugg();
            lugg.ShowDialog();
            name = first + " " + middle + " " + last;
            lstLuggage.Items.Add(name);
            writer.WriteLine(name);
            writer.Close();
            fs.Close();
        }
```

```
private void btnSecurity_Click(object sender, EventArgs e)
{
   //Create a list of passengers
   List<string> passengers = new List<string>();
   string line;
   string message;

   //Create FileStream and StreamReader to read passengers.txt
   FileStream fs = new FileStream("passengers.txt", FileMode.Open, FileAccess.Read);
   StreamReader reader = new StreamReader(fs);

   //Prime the loop
   line = reader.ReadLine();

   //Read in passengers and add to list
   while (line != null)
   {
      passengers.Add(line);
      line = reader.ReadLine();
   }

   //Close StreamReader and FileStream
   reader.Close();
   fs.Close();

   //Create FileStream and StreamReader for luggage file
   fs = new FileStream("luggage.txt", FileMode.Open, FileAccess.Read);
   reader = new StreamReader(fs);

   //Prime the loop
   line = reader.ReadLine();

   //Check luggage has a passenger
   int count = 0;
   while (line != null)
   {
      if (passengers.Contains(line))
      {
         count++;
      }
      else
      {
         lstLuggage.SelectedIndex = count;
         count++;
         message = "Security Violation - Arrest Warrant Issued or " +
            lstLuggage.SelectedItem;
         MessageBox.Show(message);
      }
```

```
                line = reader.ReadLine();
            }
            //Close StreamReader and FileStream
            reader.Close();
            fs.Close();
        }

        private void frmMain_Load(object sender, EventArgs e)
        {
            string line;
            //Create FileStream and StreamReader to read passengers.txt
            FileStream fs = new FileStream("passengers.txt", FileMode.Open, FileAccess.Read);
            StreamReader reader = new StreamReader(fs);

            //Prime the loop
            line = reader.ReadLine();

            //Read in passengers and add to list
            while (line != null)
            {
                lstPassengers.Items.Add(line);
                line = reader.ReadLine();
            }

            //Close StreamReader and FileStream
            reader.Close();
            fs.Close();

            //Create FileStream and StreamReader for luggage file
            fs = new FileStream("luggage.txt", FileMode.Open, FileAccess.Read);
            reader = new StreamReader(fs);

            //Prime the loop
            line = reader.ReadLine();

            //Check luggage has a passenger
            while (line != null)
            {
                lstLuggage.Items.Add(line);
                line = reader.ReadLine();
            }

            //Close StreamReader and FileStream
            reader.Close();
            fs.Close();
        }
    }
}
```

```
//Passenger entry form
//Author: nmessa
//Date: 6.6.2022

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 Security
{
    public partial class frmPass : Form
    {
        public frmPass()
        {
            InitializeComponent();
        }

        private void btnAddPass_Click(object sender, EventArgs e)
        {
            frmMain.first = txtFirst.Text;
            frmMain.middle = txtMiddle.Text;
            frmMain.last = txtLast.Text;
            this.Close();
        }

        private void frmPass_Load(object sender, EventArgs e)
        {
            txtFirst.Focus();
        }
    }
}
```

```
//Luggage entry form
//Author: nmessa
//Date: 6.6.2022

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 Security
{
    public partial class frmLugg : Form
    {
        public frmLugg()
        {
            InitializeComponent();
        }

        private void btnAddPass_Click(object sender, EventArgs e)
        {
            frmMain.first = txtFirst.Text;
            frmMain.middle = txtMiddle.Text;
            frmMain.last = txtLast.Text;
            this.Close();
        }

        private void frmLugg_Load(object sender, EventArgs e)
        {
            txtFirst.Focus();
        }
    }
}
```