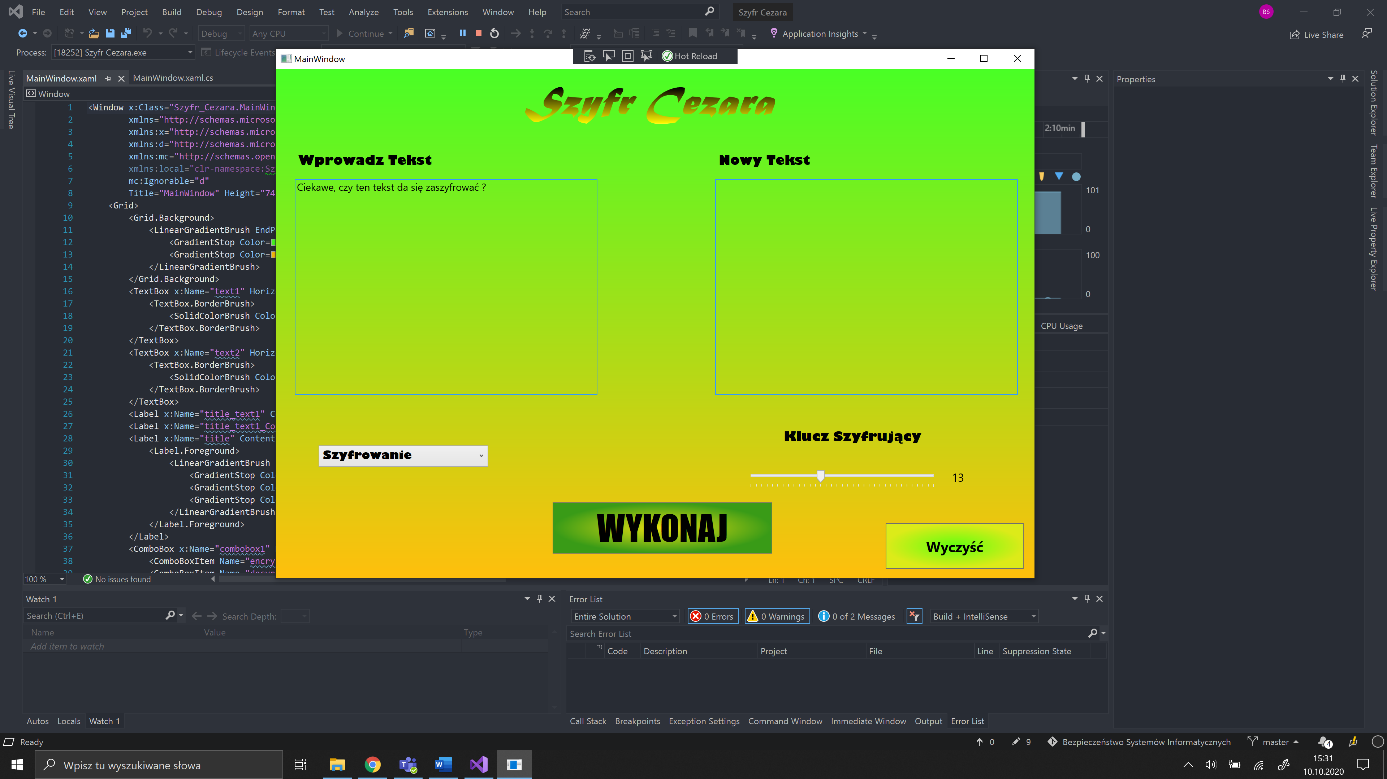
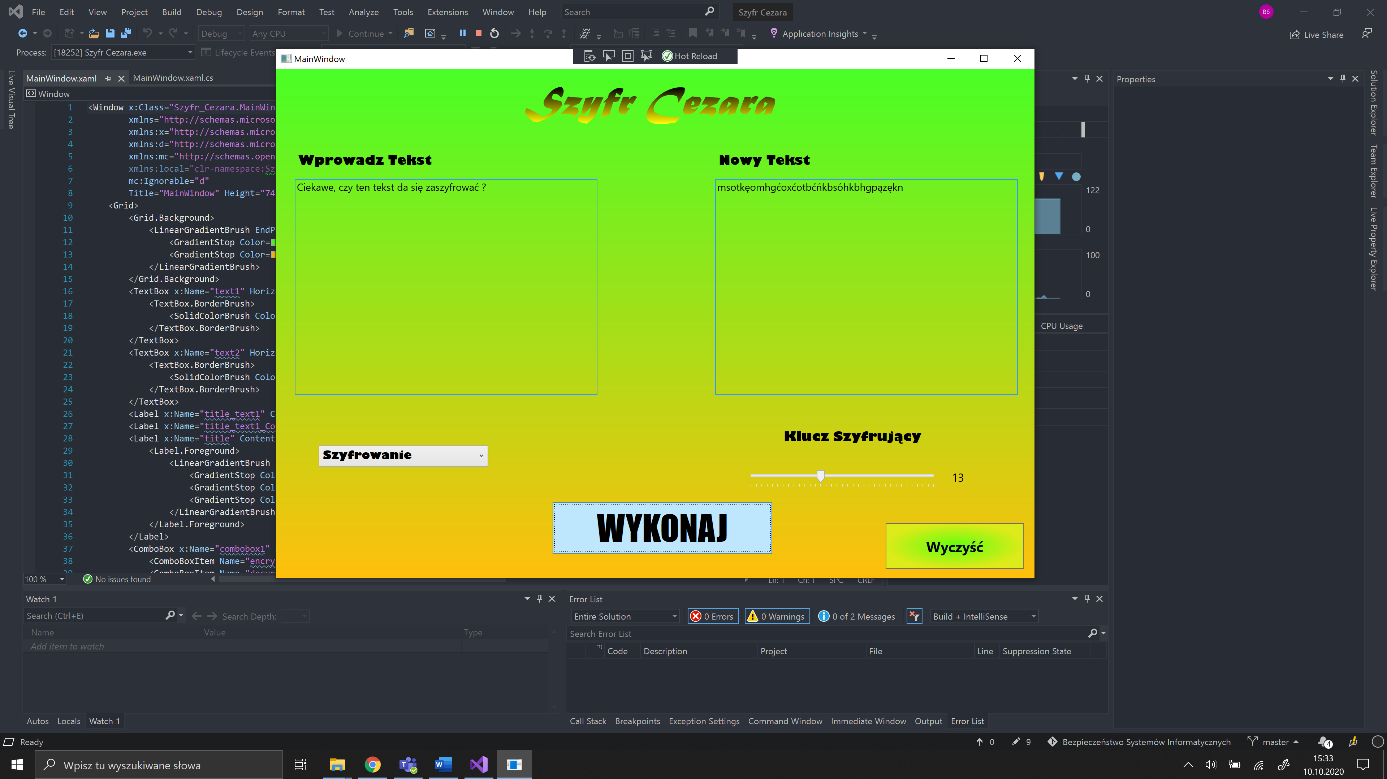
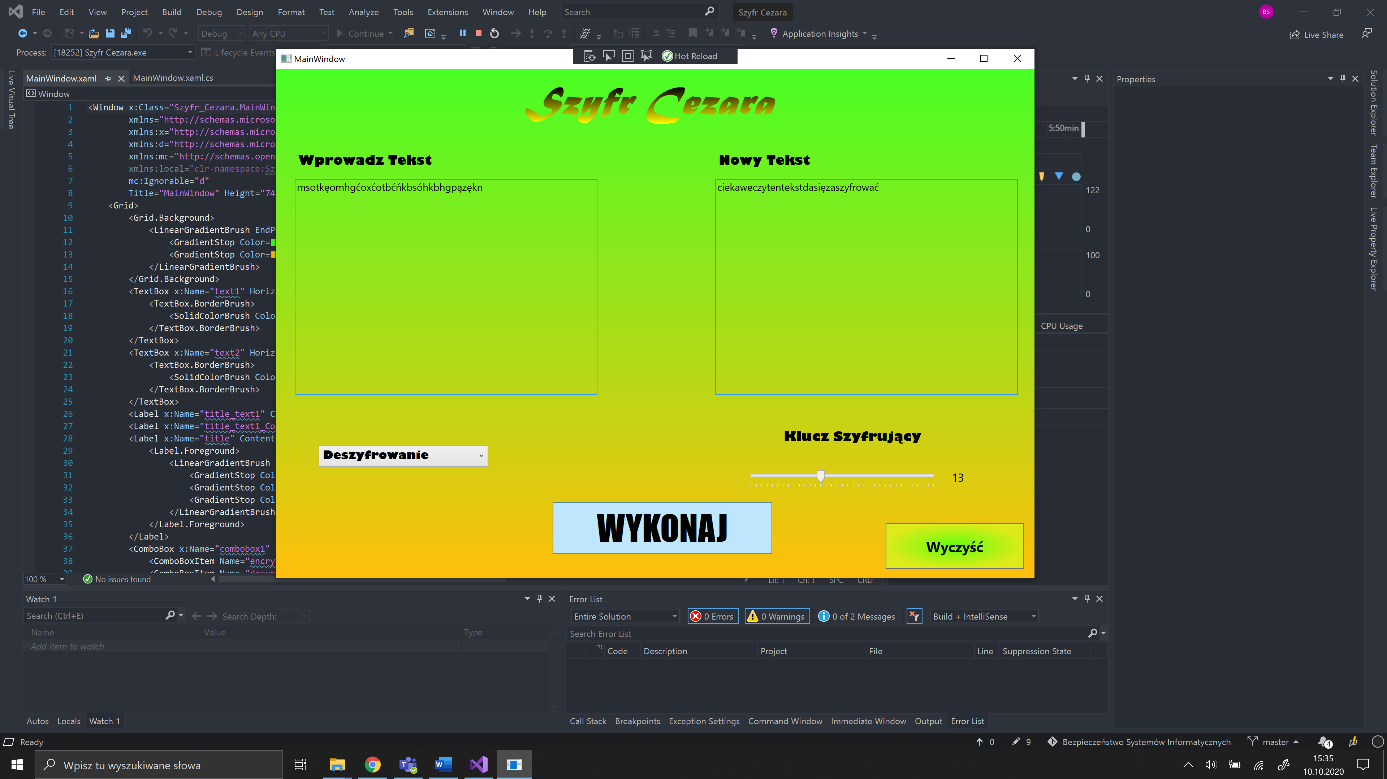
**Szyfr Cezara**

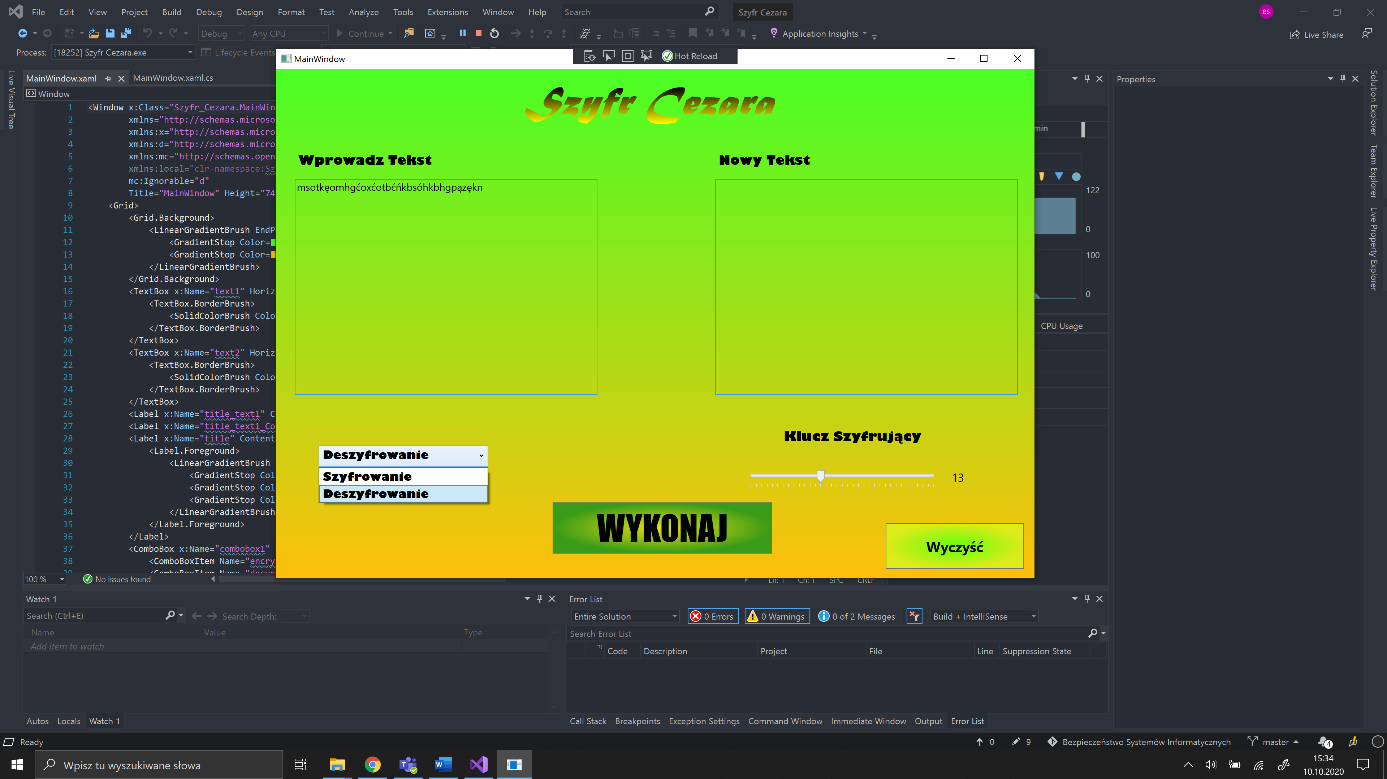
**Aplikacja okienkowa napisana w języku C#**

**Bartłomiej Szewczyk**

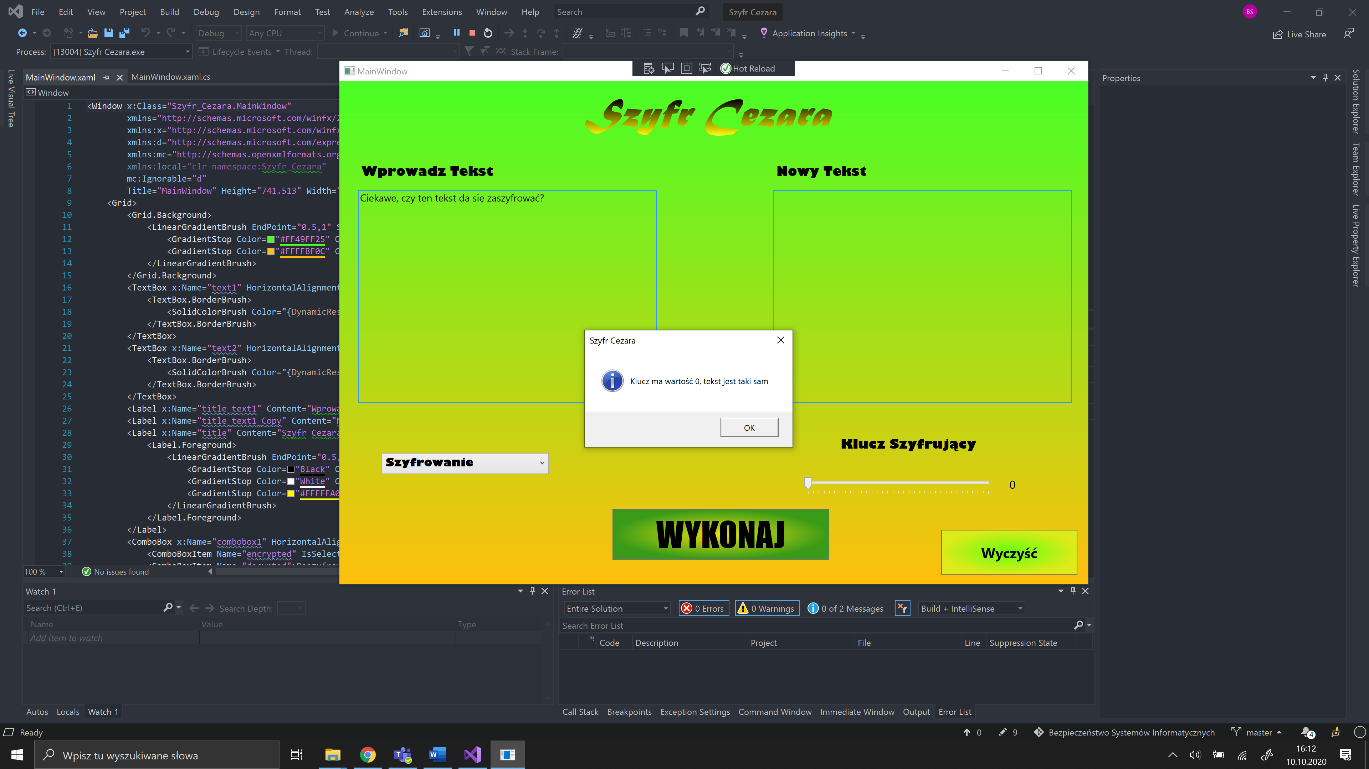


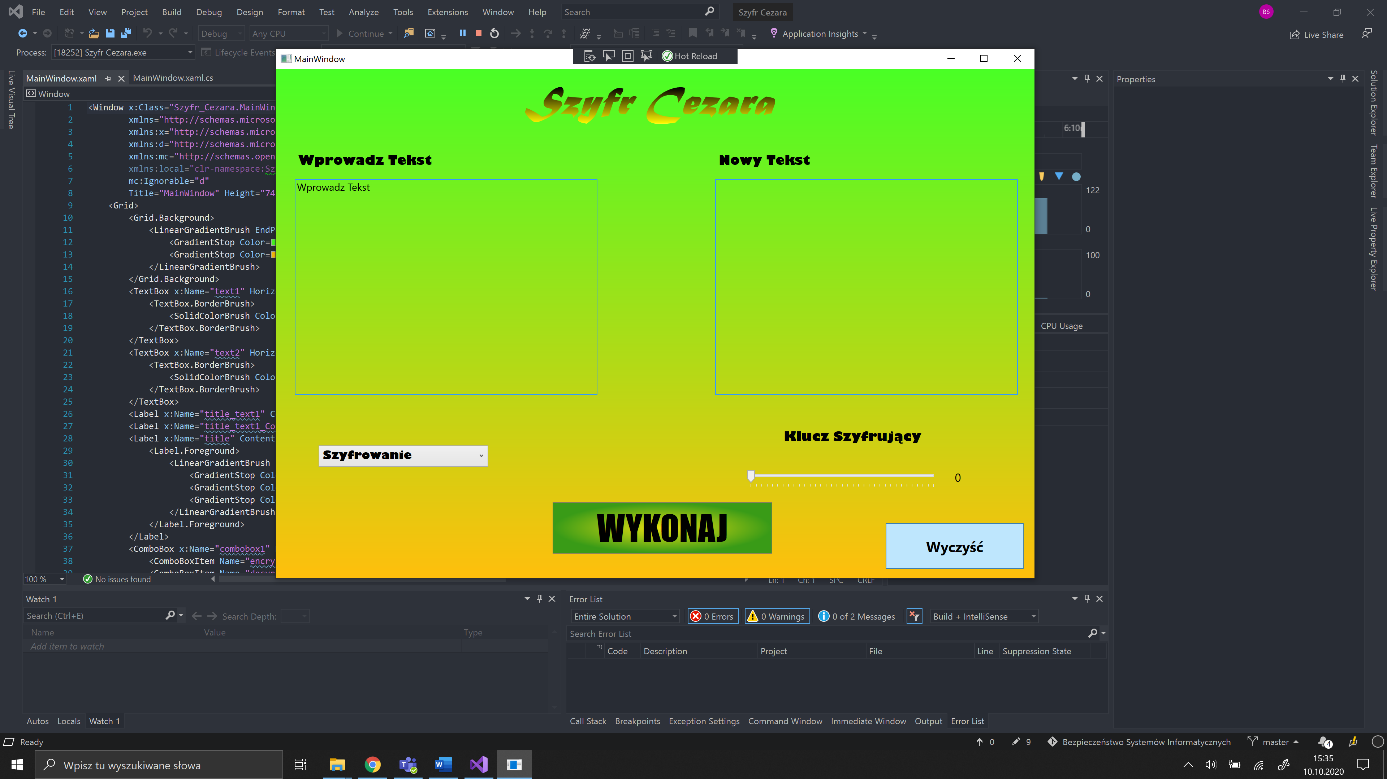
1. Aplikacja posiada 2 pola tekstowe 1 do którego wprowadzamy tekst który chcemy zaszyfrować bądź odszyfrować oraz 2 w którym wyświetla się nasz szyfr lub odszyfrowana wiadomość(to okienko nie posiada funkcji edycji tekstu by przypadkiem nie zmienić szyfru).





1. Po wybraniu polecenia oraz Klucza Szyfrującego który działa w zakresie 0-34 i naciśnięciu przycisku wykonaj tekst zostanie zaszyfrowany bądź odszyfrowany wraz z usunięciem z tekstu znaków interpunkcyjnych i spacji.



1. Podczas próby odszyfrowywania lub szyfrowania wiadomości której klucz szyfrujący jest ustawiony na 0 wyskoczy komunikat informujący o tym że nowy tekst będzie identyczny co wpisana przez nas wiadomość.
2. W każdym momencie możemy przywrócić program do stanu początkowego(wyczyścić wszystkie wartości i teksty ) naciskając przycisk „Wyczyść”.

**Kod aplikacji napisany w języku C#**

**Repozytorium GitHub:** [**https://github.com/Bartek-Szewczyk/Szyfr-Cezara-CSharp**](https://github.com/Bartek-Szewczyk/Szyfr-Cezara-CSharp)

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Szyfr\_Cezara

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

public void Szyfruj()

{

char[] alphabet = new char[] { 'a', 'ą', 'b', 'c', 'ć', 'd', 'e', 'ę', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'ł', 'm', 'n', 'ń', 'o', 'ó', 'p', 'q', 'r', 's', 'ś', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'ź', 'ż' };

int key = (int)slValue.Value;

char[] znak = new char[] { '.', ',', '?', '!', '@' };

string orginaltext = text1.Text.ToLower().Replace(" ", "").Replace(".", "").Replace(",", "").Trim(znak);

char[] secretText = orginaltext.ToCharArray();

char[] encryptedText = new char[secretText.Length];

for (int i = 0; i < secretText.Length; i++)

{

char secretItem = secretText[i];

int index = Array.IndexOf(alphabet, secretItem);

int letterPosition = (index += key) % 35;

char encriptedCharacter = alphabet[letterPosition];

encryptedText[i] = encriptedCharacter;

}

string secretTextFinal = String.Join("", encryptedText);

text2.Text = secretTextFinal;

}

public void btn1\_Click(object sender, RoutedEventArgs e)

{

if (combobox1.Text == "Szyfrowanie")

{

if (slValue.Value == 0)

{

MessageBox.Show("Klucz ma wartość 0, tekst jest taki sam", "Szyfr Cezara", MessageBoxButton.OK, MessageBoxImage.Information);

text2.Text = text1.Text;

}

else

{

Szyfruj();

}

}

else

{

if (slValue.Value == 0)

{

MessageBox.Show("Klucz ma wartość 0, tekst jest taki sam", "Szyfr Cezara", MessageBoxButton.OK, MessageBoxImage.Information);

text2.Text = text1.Text;

}

else

{

Deszyfrowanie();

}

}

}

private void Deszyfrowanie()

{

char[] alphabet = new char[] { 'ż', 'ź', 'z', 'y', 'x', 'w', 'v', 'u', 't', 'ś', 's', 'r', 'q', 'p', 'ó', 'o', 'ń', 'n', 'm', 'ł', 'l', 'k', 'j', 'i', 'h', 'g', 'f', 'ę', 'e', 'd', 'ć', 'c', 'b', 'ą', 'a' };

int key = (int)slValue.Value;

string orginaltext = text1.Text.ToLower().Replace(" ", "");

char[] secretText = orginaltext.ToCharArray();

char[] encryptedText = new char[secretText.Length];

for (int i = 0; i < secretText.Length; i++)

{

char secretItem = secretText[i];

int index = Array.IndexOf(alphabet, secretItem);

int letterPosition = (index += key) % 35;

char encriptedCharacter = alphabet[letterPosition];

encryptedText[i] = encriptedCharacter;

}

string secretTextFinal = String.Join("", encryptedText);

text2.Text = secretTextFinal;

}

private void btnclean\_Click(object sender, RoutedEventArgs e)

{

Clean();

}

private void Clean()

{

text1.Text = "Wprowadz Tekst";

text2.Text = "";

slValue.Value = 0;

encrypted.IsSelected = true;

}

}

}