Caesar Cypher

This program allows the user to input a series of strings, and then either have the sequence encrypted via a shift (up or down) of their choice, or to have a random shift of 1-10 applied. The program then displays the encrypted string and if a random shift was chosen, it will also display the shift amount, needed to decrypt the message.

Public Class frm\_Main

Dim cypherinput As String 'the string input is assigned here

Public shiftamount As Integer 'the integer input for the shift

Public cypheroutput As String 'the string after it has been shifted

Dim inputlength As Integer 'the length of the input string

Dim shiftarr(1) As Integer 'an array storing the ASCII of the shifted character

Dim arr(1) As String 'an array storing the shifted character

Dim ran As Integer 'takes the value of the randomly generated number

Dim rd As New Random 'used to generate a random number

Private Sub btn\_RandomShift\_Click(sender As Object, e As EventArgs) Handles btn\_RandomShift.Click

cypherinput = UCase(txt\_InputWord.Text) 'converts string to uppercase

inputlength = txt\_InputWord.TextLength - 1 'corrects the string length so that calculations are correct

ran = rd.Next(1, 11) 'generates a random integer between inclusive 1 and exclusive 11 (1-10)

shiftamount = ran

ReDim shiftarr(inputlength) 're-declares the shiftarr to the size of the string length

ReDim arr(inputlength) 're-declares the arr to the size of the string length

If txt\_InputWord.Text = "" Then

MsgBox("No String Input")

GoTo escape1

End If

For i = 0 To inputlength

If cypherinput(i) <> " " Then

shiftarr(i) = Asc(cypherinput(i)) + shiftamount 'converts the current character into ASCII and adds the shift amount onto the ASCII value

If shiftarr(i) > 90 Then

shiftarr(i) = shiftarr(i) - 26 'if the ASCII value is above the uppercase Z value, that particular ASCII value is moved back down to uppercase A, creating a loop like a physical caesar cypher would create

End If

arr(i) = Convert.ToChar(shiftarr(i))

ElseIf cypherinput(i) = " " Then 'spaces are kept in the same place

arr(i) = " "

End If

Next

For i = 0 To inputlength

cypheroutput = cypheroutput + arr(i) 'this re-creates the string with the new encrypted characters

Next

txt\_Display.Text = txt\_Display.Text + cypheroutput & " Shift Amount: " & shiftamount.ToString & ControlChars.NewLine 'displays the encrypted (or decrypted) string in the textbox on the bottom, and creates a new line

cypheroutput = ""

escape1:

End Sub

Private Sub btn\_positive\_Click(sender As Object, e As EventArgs) Handles btn\_Positive.Click

'majority of this code is the same as previous, with only some of the variables changed and lacking the integer generation

cypherinput = UCase(txt\_InputWord.Text)

inputlength = txt\_InputWord.TextLength - 1

Try 'a try catch block, in case there is a problem with the shift input

shiftamount = txt\_CustomShift.Text

Catch ex As Exception

MessageBox.Show("Invalid Shift Input")

End Try

If shiftamount >= 26 Or shiftamount < 1 Then

MessageBox.Show("Please input a number between 1 and 25")

GoTo escape1

End If

ReDim shiftarr(inputlength)

ReDim arr(inputlength)

For i = 0 To inputlength

If cypherinput(i) <> " " Then

shiftarr(i) = Asc(cypherinput(i)) + shiftamount

If shiftarr(i) > 90 Then

shiftarr(i) = shiftarr(i) - 26 'this accounts for the "Positive" shift

End If

arr(i) = Convert.ToChar(shiftarr(i))

ElseIf cypherinput(i) = " " Then

arr(i) = " "

End If

Next

For i = 0 To inputlength

cypheroutput = cypheroutput + arr(i)

Next

txt\_Display.Text = txt\_Display.Text + cypheroutput & ControlChars.NewLine

cypheroutput = ""

escape1:

End Sub

Private Sub txt\_customshift\_KeyPress(ByVal sender As System.Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txt\_CustomShift.KeyPress

If Not Char.IsDigit(e.KeyChar) And Not Char.IsControl(e.KeyChar) Then

e.Handled = True 'disables the user from inputting any non-integer characters

End If

End Sub

Private Sub btn\_Clear\_Click(sender As Object, e As EventArgs) Handles btn\_Clear.Click

txt\_Display.Clear() 'clears the display

End Sub

Private Sub btn\_Negative\_Click(sender As Object, e As EventArgs) Handles btn\_Negative.Click

cypherinput = UCase(txt\_InputWord.Text) 'identical to the positive shift sub, just moves the ASCII down instead of up

inputlength = txt\_InputWord.TextLength - 1

Try

shiftamount = txt\_CustomShift.Text

Catch ex As Exception

MessageBox.Show("Invalid Shift Input")

End Try

If shiftamount >= 26 Or shiftamount < 1 Then

MessageBox.Show("Please input a number between 1 and 25")

GoTo escape1

End If

ReDim shiftarr(inputlength)

ReDim arr(inputlength)

For i = 0 To inputlength

If cypherinput(i) <> " " Then

shiftarr(i) = Asc(cypherinput(i)) - shiftamount

If shiftarr(i) < 65 Then

shiftarr(i) = shiftarr(i) + 26 'this accounts for the "Negative" shift

End If

arr(i) = Convert.ToChar(shiftarr(i))

ElseIf cypherinput(i) = " " Then

arr(i) = " "

End If

Next

For i = 0 To inputlength

cypheroutput = cypheroutput + arr(i)

Next

txt\_Display.Text = txt\_Display.Text + cypheroutput & ControlChars.NewLine

cypheroutput = ""

escape1:

End Sub

Private Sub txt\_InputWord\_KeyPress(ByVal sender As System.Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txt\_InputWord.KeyPress

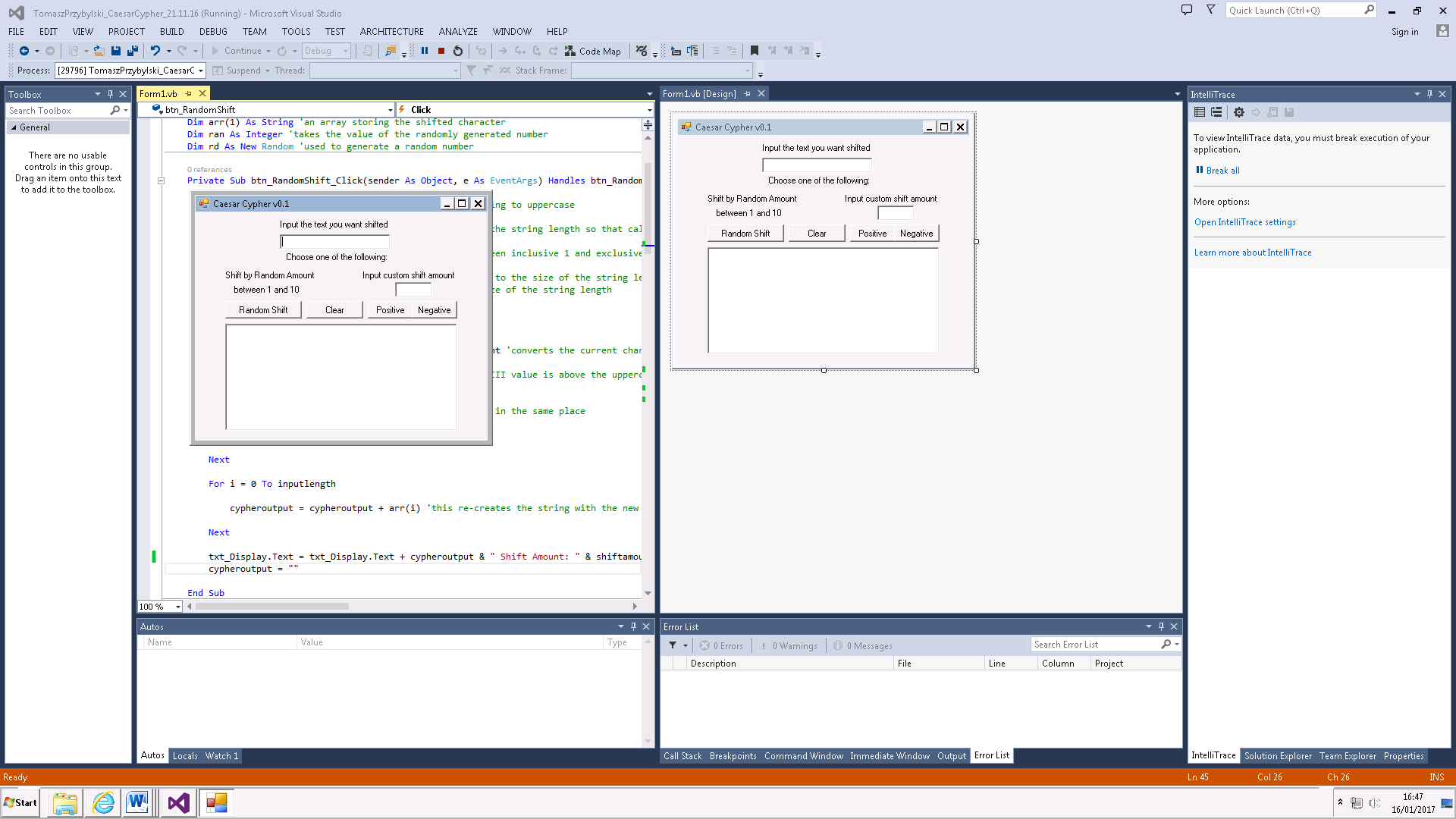
If Not Char.IsLetter(e.KeyChar) And Not Char.IsControl(e.KeyChar) And Not Char.IsSeparator(e.KeyChar) Then

e.Handled = True 'disables input of numbers of symbols (apart from Spacebar) into the "Input text" textbox

End If

End Sub

End Class



Output textbox. Will display the shifted

Shifts input text by a random amount between 1 and 10

Clears output textbox

Negative shift

Positive shift

Input for shift amount

Input for string

## Comments:

* I made two separate arrays for this program. Although the program could be created without using these and by re-creating the string during the loop, this method minimises the chances of errors or mix-ups during the code and makes it much friendlier to understand by looking at it.
* The clear button is only included for user-friendliness. It does not affect how the program executes
* Originally my program just shifted the input text either randomly or by a custom amount, but I decided to split that up to Positive and Negative shift so that the program could be also used to decrypt anything as long as the shift amount is known. This also doubles as a debug tool to check if the input text was being shifted correctly one way or another.
* I also decided to limit the shifts to letters like a real Caesar Cypher would, and in the same way that a Caesar Cypher loops with Z shifted by 1 equalling A, I made the code stay within the bounds of the ASCII letters (with the exception of a spacebar)
* I disabled any number or symbol input for the string.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | Description | Input | Expected Outcome | Actual Outcome | Other Comments |
| #1 | Input 11 Character string into textbox & shift by +11 | Hello Hello | SPWWZ SPWWZ | As expected |  |
| #2 | Input 11 Character string into textbox & shift by -11 | Hello Hello | WTAAD WTAAD | As expected |  |
| #3 | Input 20 character string into textbox & shift by +- 13 | ABCDEFGHIJKLMNOPQRST | NOPQRSTUVWXYZABCDEFG | As expected |  |
| #4 | Input numbers & symbols into textbox | 1234+-{} | Inputs not allowed to be keyed in | As expected |  |
| #5 | Shift amount set to -3 | -3 | Invalid amount message displayed | As expected |  |
| #6 | Shift amount set to 40 | 40 | Invalid amount message displayed | As expected |  |
| #7 | Random Shift used with string in textbox | Help | OLSW Shift amount: 6 | As expected |  |
| #8 | No input for string & shift attempted |  | Nothing happens | As expected |  |
| #9 | Input for string present, no shift input | ABC | Invalid shift input & error message displayed | As expected |  |
| #10 | No input for string & random shift attempted |  | No string message displayed | As expected |  |
|  |  |  |  |  |  |