Objectives

Program Purpose

- Count the number of words and alphabetic characters in a sentence.

Learning Goals

- To demonstrate use of simple FOR-NEXT loops
- To use the functions MID, LEN and TRIM
- To develop the concept of string handling introduced in Project 11 – String Handling 1
- Use of a variable array
- Use of parameter values

Design Notes

Two types of arrays are used in Visual Basic. They are both used for storing and processing large sets of similar values. ‘Control Arrays’ are sets of objects used on the interface or form. ‘Variable Arrays’ are arrays of variables used in memory for storing values. Arrays work very well with loops. This is because a loop can very quickly process all elements of an array using the index number. Arrays are declared in a similar fashion to normal variables, but their range must also be declared. For example, the code ‘Dim IntegerArray (1 to 13) as Integer’ declares an array of 14 integers. ‘Dim Addresses(20) as String’ declares an array of 21 strings values (in this case, to store addresses). Remember that 0 is the default first member of an array unless declared otherwise.

Interface

Create the interface as shown below.
Use 3 command buttons, 1 text box, 1 listbox and 7 labels.
Names of Objects

<table>
<thead>
<tr>
<th>Type of Object</th>
<th>Number</th>
<th>Names of Objects</th>
<th>Simple Initial Properties of Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form</td>
<td>1</td>
<td>frmTextProcessor</td>
<td>Caption – “Text Processor 2”</td>
</tr>
<tr>
<td>Text Box</td>
<td>1</td>
<td>txtInput</td>
<td>Font – Bold, 10</td>
</tr>
<tr>
<td>ListBox</td>
<td>1</td>
<td>lstLetters</td>
<td>Font – Bold, 10</td>
</tr>
<tr>
<td>Labels</td>
<td>8</td>
<td>Label1, Label2, Label3, Label4, Label5, lblChars, lblWords, lblSpaces</td>
<td>Font – Bold, 12 Captions – as per Interface BorderStyle- Single where a value is to be displayed and changed</td>
</tr>
</tbody>
</table>

Further Initial Properties of Objects

<table>
<thead>
<tr>
<th>Object</th>
<th>Property</th>
<th>Initial Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>frmTextProcessor</td>
<td>Startup Position</td>
<td>2 – Center Screen</td>
</tr>
<tr>
<td>txtInput</td>
<td>Text</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>LstLetter</td>
<td>Multiline</td>
<td>True</td>
</tr>
<tr>
<td>Lbl1</td>
<td>Caption</td>
<td>Type in a Sentence</td>
</tr>
<tr>
<td>Lbl2</td>
<td>Caption</td>
<td>Number of Words</td>
</tr>
<tr>
<td>Lbl3</td>
<td>Caption</td>
<td>Number of Letters</td>
</tr>
<tr>
<td>Lbl4</td>
<td>Caption</td>
<td>Number of Other Characters</td>
</tr>
<tr>
<td>Lbl5</td>
<td>Caption</td>
<td>Summary of Letters and Tally</td>
</tr>
</tbody>
</table>

Events – Code

```
Dim Letters(1 To 26) As Integer 'stores an array of 26 letters  (PUT IN GENERAL SECTION)

Private Sub Form_Load()
    txtInput.Text = "The quick brown fox jumps over the lazy dog"
End Sub

Private Sub cmdClear_Click()
    lblChars.Caption = ""
    lblWords.Caption = ""
    txtInput.Text = ""
    ClearArray
End Sub

Private Sub cmdExit_Click()
    End
End Sub
```
Private Sub cmdCountLetters_Click()

'Actually it counts characters.

Dim sentence As String
Dim CountChar As Integer, CountWords As Integer, CountSpace As Integer, Letter As Integer
Dim Char As String
ClearArray
CountChar = 0
CountSpace = 0
sentence = txtInput.Text

If Len(sentence) > 0 Then

   CountWords = 1
   For Letter = 1 To Len(sentence)
      Char = Mid(sentence, Letter, 1)
      'MsgBox Char & " = Ascii Code " & Asc(Char)
      AddtoArray Asc(Char)
      If Mid(sentence, Letter, 1) <> " " Then
         CountChar = CountChar + 1           'add 1 to character count
      Else
         CountSpace = CountSpace + 1
         CountWords = CountWords + 1
      End If
   Next Letter
   Else
      CountChar = 0
   End If

   lblChars.Caption = CountChar
   lblSpaces.Caption = CountSpace
   lblWords.Caption = CountWords
   DisplayArrayInListBox

End Sub

Sub AddtoArray(LetterCode As Integer)  ' (PUT IN GENERAL SECTION)

'convert LetterCode to 1 to 26

Select Case LetterCode
Case 65 To 90  'Letter is lowercase - 65 to 90
   LetterCode = LetterCode - 64
   Letters(LetterCode) = Letters(LetterCode) + 1
Case 97 To 122 'letter is uppercase - 97 to 122
   LetterCode = LetterCode - 96
   Letters(LetterCode) = Letters(LetterCode) + 1
Case Else
End Select
'MsgBox LetterCode

End Sub

Sub DisplayArrayInListBox()  ' (PUT IN GENERAL SECTION)

Dim i As Integer
lstLetters.Clear

For i = 1 To 26
   lstLetters.AddItem Chr(i + 96) & "  " & Letters(i)
Next i

End Sub
Sub ClearArray() 'PUT IN GENERAL SECTION

    Dim i As Integer
    For i = 1 To 26
        Letters(i) = 0
    Next i

End Sub

Private Sub txtInput_Change()
    cmdCountLetters.Value = True
End Sub

Private Sub txtInput_Change()
    cmdCountWords.Value = True
    cmdCountLetters.Value = True
End Sub

Notes

Asc (character) will return the ASCII value of any character.

Chr (ascii value between 1 and 128) will change an ASCII value into a character.

Suggestions for Consolidation and Extension

1. Modify the program by adding another listbox called 'lstChars'. It should work like the 'lstletters' listbox and display a tally of non-alphanumeric characters. You will need a new array declaration as well.

2. Create a program to display all ASCII characters and their corresponding values. Print it out and keep it for reference.

Questions

1. What outputs do ‘CHR(82)’ and ‘CHR(43)’ return?

2. What outputs do ‘ASC(‘K’)’ and ‘ASC(‘k’)’ return?

3. What outputs do ‘CHR(10)’ and ‘CHR (13)’ return?

4. Why declare “Dim Letters (1 to 26) as Integer” and not simply ”Dim letters(26) as Integer”?

5. Is there any redundancy in the ‘String Handling 1’ program (redundancy is extra code or objects on the interface that are not needed)?

6. How can we solve the logical error of counting a space as a word and counting two spaces in a row as two words?

7. What other enhancements and modifications might be useful for these programs?