Please answer all the following questions on this test document. Question point values are shown in parentheses.

1. (14) In class we looked at a general sub procedure that inserted dashes into a social security number that was similar to the following:

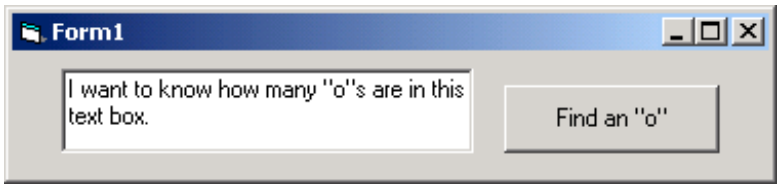
```
Private Sub insertDashes()
   If Len(ssn) <> 9 Then
      MsgBox "SSN must be 9 digits in length"
      Exit Sub
   End If
   ssn = Left$(ssn, 3) & "-" & Mid$(ssn, 4, 2) & "-" & Right$(ssn, 4)
End Sub
```

   We also looked at a general function procedure that essentially did the same thing. It was similar to the following:

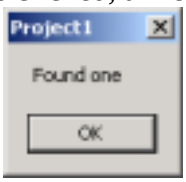
```
Private Function insertTheDashes(ByVal s As String) As String
   If Len(s) <> 9 Then
      insertTheDashes = "X"
      Exit Function
   End If
   insertTheDashes = Left$(s, 3) & "-" & Mid$(s, 4, 2) & "-" & Right$(s, 4)
End Function
```

   Identify which solution approach is superior and indicate why you come to this conclusion.

2. (15) You are given the following user interface:

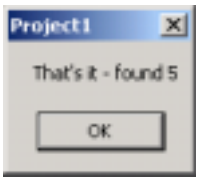


The text box on the left is named txtWords. The first time the command button on the right is clicked, a message box like the following should be displayed:



The second and subsequent times the command button is clicked and there is another "o" found, a message box like the following should be displayed:

When the command button is clicked and no more "o"s can be found, a final message like the following should be displayed:



Write the logic for the click event whose behavior is the same as described above. Use minimal scope for full credit. Your solution should work for any text.

3. (15) Write a KeyPress event for a text box named txtSerialNo that enforces the following rules:

Characters 1 and 2:	Digits 0 – 9 only.
Character 3:	Any letter (A – Z or a – z). If lowercase is entered, change to uppercase before display.
Character 4:	A dash (-) (no other character is legal here).
Character 5:	An uppercase X (no other character is legal here).

There cannot any characters beyond the 5th character. If an illegal character is detected in any of the positions, have the computer "beep" and reject the illegal character. A backspace should be allowed at any time.

4. (14) What output is displayed by the following code when cmdStartHere is clicked?

```
' form's General Declaration Section

Option Explicit
Dim a As Integer, b As Integer, c As Integer


Private Sub cmdStartHere_Click()
    Dim d As Integer
    a = 10
    b = 20
    apple a, b, c, d
    Form1.Print a, b, c, d  'what is printed
       here?
    cherry a, b, d
    Form1.Print a, b, d  'what is printed here?
End Sub
```

```
Private Sub apple(a As Integer, b As
      Integer, ByVal c As Integer, d As
      Integer)
    c = a
    d = b
    a = 100
    b = 200
End Sub

Private Function banana(ByVal a As
      Integer, ByVal b As Integer) As
      Integer
    Dim t As Integer
    t = a
    a = c
    c = t
    b = 300
End Function

Private Sub cherry(x As Integer, y As
      Integer, z As Integer)
    Dim a As Integer, b As Integer, c As
      Integer
    a = x
    b = banana(y, z)
    y = z
    x = a * b
End Sub
```

5. (14). You are given the following statements:

```
Dim a As Integer, b As Integer, c As Integer, d As Integer
Dim s As String, t As String
Dim x As Boolean, y As Boolean, z As Boolean
a = 10: b = 30: c = 50: d = 2
s = "Seattle": t = "battle"
x = True: y = False: z = False
```

Determine the value of each expression below given the initial values above.

a. `c \ b`

b. `c Mod b`

c. `d + a / d + b`

d. `-a ^ d`

e. `Mid$(s, 3, 5) = Right$(t, 5)`

f. `x And y Or z`

g. `InStr(1, s, "at") Mod 2`

6. (14) For each of the following statements indicate if it is true or false.

a. The portability of code can be improved by limiting the use of local scope and using module-level scope for variables instead.

b. It is considered “safer” and therefore preferable to use “ByVal” parameters in general function procedures.

c. If a general sub procedure needs to modify a value declared in the calling procedure then you can either use global scope or a “ByRef” parameter.

d. When comparing two string values, shorter strings are always “less than” longer strings.

e. If you place two or more option buttons into a frame container then they will be treated as a group and only one can be selected at a time.

f. If the Value property of a check box component is true then the check box will show a checkmark in it.

g. The keyword Static extends the scope of a local variable to be the equivalent of module-level scope.

7. (14) Write a segment of code (not an entire procedure) to determine if the values in two text boxes (txtOne and txtTwo) are >, < or = to each other. Use a message box to display the results of the comparison. Assume that each text box contains a numeric value.

   Be sure that the comparison is made using a numeric relationship, for example, 2 < 11 should be evaluated as true.

1. The second solution (the function) is superior because it is more portable and modular plus it uses narrower and saver scope.

   It's more portable and modular because it does not require the application that uses it to define a broadly scoped variable named "ssn". Instead, it uses arguments and parameters to share a variable in a read-only manner.

   It is also more portable because it does not impose a message box on the user of the function – rather it just finds the problem (incorrect number of characters) and reports that condition (leaving the decision on what to do in response to the error up to the using application).

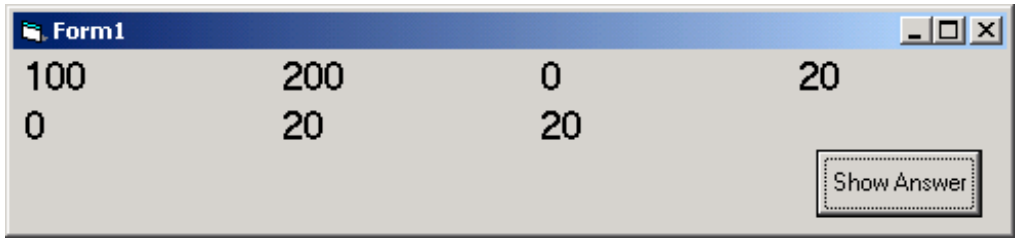
2. One possible solution is:

```
Private Sub cmdFindOh_Click()
   Static locOfAnO As Integer
   Static countOfO As Integer
   locOfAnO = InStr(locOfAnO + 1, txtWords.Text, "o")
   If locOfAnO = 0 Then
      MsgBox "That's it - found " & countOfO
      Exit Sub
   End If
   If countOfO = 0 Then
      MsgBox "Found one"
   Else
      MsgBox "Found another"
   End If
   countOfO = countOfO + 1
End Sub
```

3. A solution is:

```
Private Sub txtSerialNo_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyBack Then Exit Sub
    Select Case Len(txtSerialNo.Text)
        Case 0, 1
            If KeyAscii >= vbKey0 And KeyAscii <= vbKey9 Then Exit Sub
            KeyAscii = 0
            Beep
        Case 2
            If KeyAscii >= vbKeyA And KeyAscii <= vbKeyZ Then Exit Sub
            If KeyAscii >= Asc("a") And KeyAscii <= Asc("z") Then
                KeyAscii = KeyAscii - 32
            Else
                KeyAscii = 0
                Beep
            End If
        Case 3
            If KeyAscii <> Asc("-") Then
                KeyAscii = 0
                Beep
            End If
        Case 4
            If KeyAscii <> vbKeyX Then
                KeyAscii = 0
                Beep
            End If
        Case Else
            KeyAscii = 0
            Beep
        End Select
End Sub
```

4. The output is:



5. a. 1

   b. 20

   c. 37

   d. -100

   e. True

   f. False

   g. 1

6. a. False. Just the opposite is true.

   b. True.

   c. False. A variable declared in the calling procedure has local scope by definition.

   d. False. True only if they contain identical characters when the end of the shorter one is detected.

   e. True

   f. False. The Value property of a check box is not Boolean.

   g. False. Static extends the lifetime, not the scope of a local variable.

7. One possible solution is:

```
If Val(txtOne.Text) > Val(txtTwo.Text) Then
   MsgBox "txtOne > txtTwo"
ElseIf Val(txtOne.Text) < Val(txtTwo.Text) Then
   MsgBox "txtOne < txtTwo"
Else
   MsgBox "txtOne = txtTwo"
End If
```