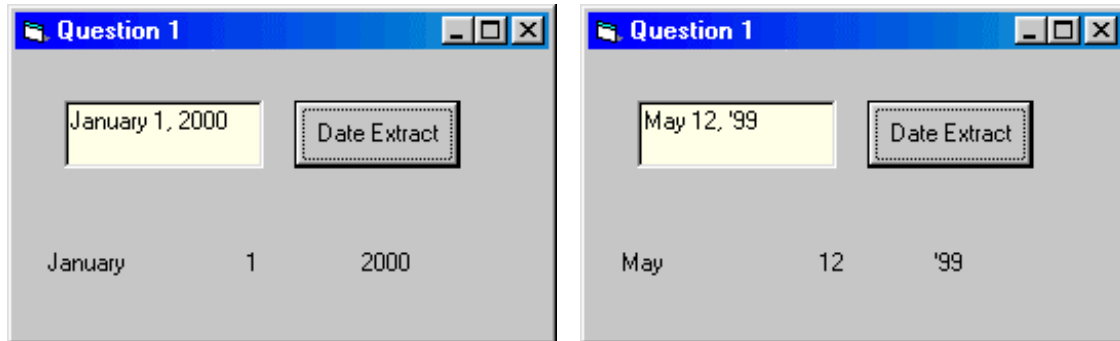


Please answer **all** the following questions on **separate sheets of paper**. Question point values are shown in parentheses.

1. (16) You are to write the code that takes a string formatted as

month day, year

and extracts the individual components (the month, the day, and the year). For example, consider the following two screen images.



The following incomplete code segment extracts the month, day, and year from a string stored in a text box named txtDate.

```
Private Sub cmdDateExtract_Click()  
    Dim wholeDate As String  
    Dim month As String  
    Dim day As String  
    Dim year As String  
    ' Dim additional variables if necessary  
    wholeDate = txtDate.Text  
  
    ' Fill in the code to perform the extract  
  
    lblMonth.Caption = month  
    lblDay.Caption = day  
    lblYear.Caption = year  
End Sub
```

Fill in the missing code so that the variables month, day, and year are defined as described above. You may assume that the original date is formatted correctly. That is, assume that a space exists between the month and day, and a comma follows the day.

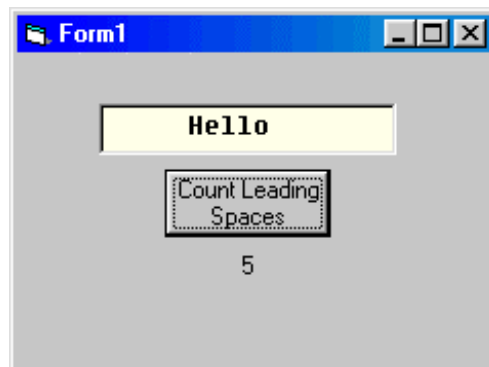
2. (12) Consider the following VB code:

```
Private Sub cmdQuest2_Click()  
    Dim a As Integer  
    Dim b As Integer  
    Dim c As Double  
    Dim d As Double  
    Dim x As Double  
  
    a = 20  
    b = 10  
    c = 2.5  
    d = 0.5  
  
    x = c / d * b \ a  
    Form1.Print x           ' Part a  
  
    x = c * b Mod a + b  
    Form1.Print x           ' Part b  
  
    x = -b ^ 2 / -d  
    Form1.Print x           ' Part c  
  
    x = (a \ b + a / b) * b Mod a  
    Form1.Print x           ' Part d  
  
End Sub
```

What value of the variable x is printed for each part (a, b, c, and d)?

3. (14) Considering parameters for functions and procedures, would you be more likely to use “by reference” for the parameters in a procedure and “by value (or ByVal)” for parameters in a function? Explain.

4. (14) The screen image below shows an application that counts the number of blank spaces that precede the first nonblank character in the text box.



The command button's click event code follows:

```
Private Sub cmdCountLeading_Click()  
    Dim phrase As String  
    Dim count As Integer  
    phrase = txtWord.Text  
    count = countOfLeadingSpaces(phrase) ' write this procedure  
    lblCount.Caption = count  
End Sub
```

Write the **complete** procedure “countOfLeadingSpaces” that is called in the code above.

5. (12) Identify each of the following statements as being either True or False.
- The lifetime of a local variable can never be the same as the lifetime of a module-level variable.
 - A variable displayed in a label control will be preceded with a dollar sign if the variable has been declared with the Currency data type.
 - The IIF function and the If statement are functionally equivalent.
 - The following Select Case statement is syntactically valid.

```
Select Case x  
    Case x < 0 Or x > 10  
        ' whatever  
    Case Else  
        ' whatever else  
End Select
```

6. (18) You are given the following click event and three sub procedures

<pre>Option Explicit ' general declaration section Dim a As Integer</pre>	<pre>Public Sub subA(x As Integer, y As Integer, z As Integer) a = x + y y = 30 z = z + 1 End Sub</pre>
<pre>Private Sub cmdStart_Click() Dim b As Integer Dim c As Integer a = 100 b = 20 subA a, b, c Form1.Print a, b, c 'What is printed here? subB a, b, c Form1.Print a, b, c 'What is printed here? subC a, b, c Form1.Print a, b, c 'What is printed here? End Sub</pre>	<pre>Public Sub subB(ByVal x As Integer, b As Integer, c As Integer) Dim y As Integer x = b b = a y = c End Sub Public Sub subC(a As Integer, b As Integer, c As Integer) a = 50 b = 60 c = 70 subA a, b, c subB a, b, c End Sub</pre>

Determine the values of the variables a, b, and c for each of the Form1.Print statements when the cmdStart_Click event is executed.

7. (14) Assume that you have a text box named txtIncome. The user enters an income figure in this text box and you need to set an integer variable called incomeBracket equal to one of three values based on the following table.

<u>Value User Enters</u>	<u>incomeBracket Value</u>
0 to <= 15000	1
>15000 to <= 65000	2
> 65000	3

If the value entered by the user is not a valid number or is less than zero, a message box should be displayed telling the user that the income value is not valid and the incomeBracket variable should be set to -1.

Write a code **segment** for a click event that fulfills the requirements outlined above.

Answer Key

1. The following is an example of code that accomplishes the stated goal:

```
Dim locSpace As Integer
Dim locComma As Integer

locSpace = InStr(1, wholeDate, " ")
locComma = InStr(1, wholeDate, ",")

month = Mid$(wholeDate, 1, locSpace - 1)
day = Mid$(wholeDate, locSpace + 1, locComma - locSpace - 1)
year = Mid$(wholeDate, locComma + 1, Len(wholeDate) - locComma)
```

2. The values printed are:

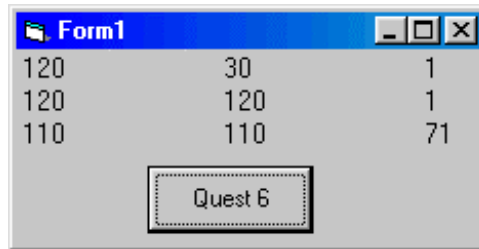


3. The answer is yes. You would be more likely to use “by reference” with parameters in a procedure because this is the only way a procedure can “return” a value to the calling procedure (short of extending variable scope which is not a good strategy). On the other hand, a function can return a value using the function name and thus is less likely to use “by reference” parameters. A function will more likely use the safer “by value” parameter type.
4. An example solution for the countOfLeadingSpaces function is:

```
Private Function countOfLeadingSpaces(ByVal w As String) As Integer
    Dim tempPhrase As String
    tempPhrase = LTrim$(w) ' remove left (leading) spaces
    countOfLeadingSpaces = Len(w) - Len(tempPhrase)
End Function
```

5. a. False
b. False
c. False
d. False

6. The values printed by the click event procedure are:



7. A code segment that fulfills the requirements of question 7 is:

```
Dim income As Long
```

```
If Not IsNumeric(txtIncome.Text) Then  
    incomeBracket = -1  
    MsgBox "Income value is not valid"  
    Exit Sub  
End If
```

```
income = txtIncome.Text  
Select Case income  
    Case 0 To 15000  
        incomeBracket = 1  
    Case 15000 To 65000  
        incomeBracket = 2  
    Case Is > 65000  
        incomeBracket = 3  
    Case Else  
        incomeBracket = -1  
        MsgBox "Income value is not valid"  
End Select
```