Name: ____________________, ________ _________
(last) (first)

Student Number: ______________________________

Section: ____ Instructor: _P. Cribb _ L. Lowther _(circle)

York University
Faculty of Science and Engineering
Department of Computer Science and Engineering

Final Exam
7:00- 10:00pm, April 14, 2008

COSC1530.03 - Introduction to Computer Use II – Winter term, FW07

Instructions:

1. Examination rules are in effect.
2. Fill in the information requested at the top of this page and print your name at the top of all the other pages.
3. Answer ALL questions. Answer questions in the space provided.
4. Time allowed is 180 minutes.
5. Use of calculators is NOT permitted.
6. Including the cover there are 19 pages with parts A to H. Please check the pages.

	Value	Mark
A.	10	_____
B.	10	_____
C.	22	_____
D.	12	_____
E.	9	_____
F.	20	_____
G.	10	_____
H.	7	_____
Total:	100	_____

Question A (10 points)

1. In the code that implements a function you may not call another function or sub.
… … T F

2. A function (call) can be used where ever a variable may be used because the function (call) essentially represents a value, i.e. the answer from the function.
… … T F

3. A formal argument that is declared ByVal may not be assigned a new value within the subprogram.
… … T F

4. The value of an actual argument corresponding to a formal argument that is declared ByRef may have been changed when the subprogram finishes executing.
… … T F

5. A subprogram that has a ByRef formal argument may not be called with an explicit value as the corresponding actual argument, e.g. Call aSub(4.5) is not valid if the subprogram definition is Private Sub aSub (ByRef x As Double)
… … T F

6. Coupling between a subprogram and the program that calls it may be through arguments, global variables, and/or control objects.
… … T F

7. It is best practice if coupling between a subprogram and the program that calls it is only through the argument list.
… … T F

8. If a subprogram includes a declaration of a local variable that is the same name as a global (form level) variable then that global variable may have been assigned a new value when the subprogram has finished executing.
… … T F

9. Selecting a file using the OpenFileDialog control actually opens and reads the file.
… … T F

10. If a file is read line-by-line generally a conditional loop will be used.
… … T F

Question B (10 points) Circle ***one*** answer that best answers each question.

1) An action that may be taken by the user is a(n) __________.

A) class B) control
C) event D) method
E) object F) procedure

2) An object used in a Graphical User Interface (GUI) is a(n) __________.

A) class B) control
C) event D) method
E) object F) procedure

3) An occurrence of a class is a(n) __________.

A) class B) control
C) event D) method
E) object F) procedure

4) A prototype or blueprint for an object is a(n) __________.

A) class B) control
C) event D) method
E) object F) procedure

5) The term ***actual parameter*** is another name for __________.

A) event B) member
C) property D) control
E) argument F) class

6) What is a sequence of instructions written to perform a specified task?

A) procedure subprogram B) program
C) programming language D) semantics
E) syntax F) function subprogram

7) What is a sequence of instructions that returns a value?

A) procedure subprogram B) program
C) programming language D) semantics
E) syntax F) function subprogram

8) What is a sequence of instructions that performs an action?

A) procedure subprogram B) program
C) programming language D) semantics
E) syntax F) function subprogram

9) Which of the following best describes the ***meaning*** of instructions written in a programming language (choose one answer):

A) file B) program
C) semantics D) class
E) syntax F) event

10) Which of the following best describes the ***formal rules*** governing the construction of valid instructions written in a programming language (choose one answer):

A) file	B) program
C) semantics	D) class
E) syntax	F) event

Question C (22 points)

To receive partial marks write *brief* notes indicating your thinking.

1. (2 points) What is displayed in the listbox named lstOutput after the following statements execute?

```
Dim s1, s2 As String                              Notes
Dim i, indx As Integer
s1 = "abcde"
s2 = "sample text string"
For i = 1 To s1.Length
   If s2.IndexOf(s1.Substring(i-1, 1)) > 0 Then
      indx = lstOutput.Items.Add("yes")
   Else
      indx = lstOutput.Items.Add("no")
   End If
Next
```

2. (2 points) What is displayed in the listbox after the following statements execute?

```
Dim aStr, bStr As String                          Notes
Dim p, indx As Integer
aStr = "A sample line of text"
p = aStr.IndexOf(" ")
Do While p >= 0
    bStr = aStr.Substring(0, p)
    aStr = aStr.Substring(p + 1)
    p = aStr.IndexOf(" ")
    indx = lstOutput.Items.Add(bStr)
Loop
```

3. (2 points) What is displayed in the listbox after the following statements execute?

```
Dim aStr, bStr As String            Notes
Dim i, indx As Integer
aStr = "naive"
For i = aStr.Length To 1 Step -1
   bStr = bStr & aStr.Substring(i-1, 1)
Next
indx = lstOutput.Items.Add(bStr)
```

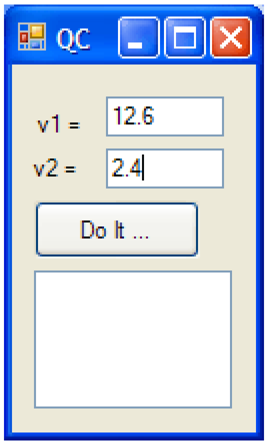
4. (2 points) Explain what is wrong with the following loop.

```
Dim q As Single
q = 1
Do While q > 0
   q = 2 * q - 1
Loop
```

5. (2 points) What is displayed in the listbox after the following statements execute?

```
Dim w As Single, indx as Integer          Notes
w = 2
Do While w < 15
    w = 2 * w - 1
   indx = lstOutput.Items.Add (w)
Loop
```

6. (2 points) Write down what is displayed in the listbox after the btnDoIt_Click sub in the program shown executes.

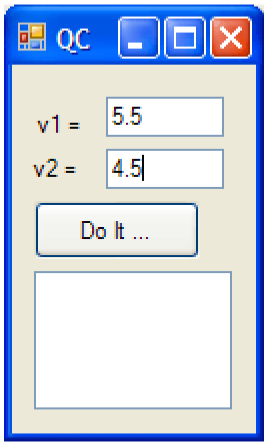


```
Public Class QC_6
    Private Sub aSub(ByVal x As Double, _
                       ByVal y As Double)
        Dim indx As Integer
        x = x + y
        indx = ListBox1.Items.Add("x = " & CStr(x))
    End Sub
    Private Sub btnDoIt_Click( ... ) _
                              Handles btnDoIt.Click
        Dim v1, v2 As Double, indx As Integer
        v1 = Convert.ToDouble(txtV1.Text)
        v2 = Convert.ToDouble(txtV2.Text)
        Call aSub(v1, v2)
        indx = ListBox1.Items.Add("v1 = " & CStr(v1))
    End Sub
End Class
```

Answer: ________________
____
____
____

Notes

7. (2 points) Write down what is displayed in the listbox after the btnDoIt_Click sub in the program shown executes.

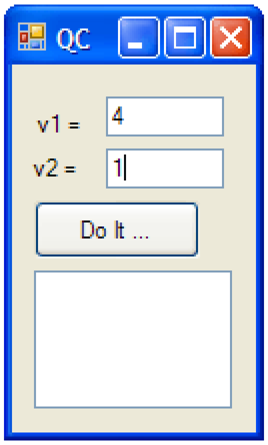


```
Public Class QC_7
    Private Sub aSub(ByVal x As Double, _
                     ByRef y As Double)
        Dim indx As Integer
        y = x + y
        indx = ListBox1.Items.Add("y = " & CStr(y))
    End Sub
    Private Sub btnDoIt_Click( ... ) _
                              Handles btnDoIt.Click
        Dim v1, v2 As Double, indx As Integer
        v1 = Convert.ToDouble(txtV1.Text)
        v2 = Convert.ToDouble(txtV2.Text)
        Call aSub(v1, v2)
        indx = ListBox1.Items.Add("v2 = " & CStr(v2))
    End Sub
End Class
```

Answer: ________________
____
____
____

Notes

8. (2 points) Write down what is displayed in the listbox after the btnDoIt_Click sub in the program shown executes.



```
Public Class QC_8
    Private x As Double, indx As Integer
    Private Sub aSub(ByVal x As Double, _
                     ByVal y As Double)
        x = x + y
        indx = ListBox1.Items.Add("x = " & CStr(x))
    End Sub
    Private Sub btnDoIt_Click( ... ) _
                             Handles btnDoIt.Click
        Dim v1, v2 As Double
        v1 = Convert.ToDouble(txtV1.Text)
        v2 = Convert.ToDouble(txtV2.Text)
        x = 3
        Call aSub(v1, v2)
        indx = ListBox1.Items.Add("x = " & CStr(x))
    End Sub
End Class
```

Answer: ____________________

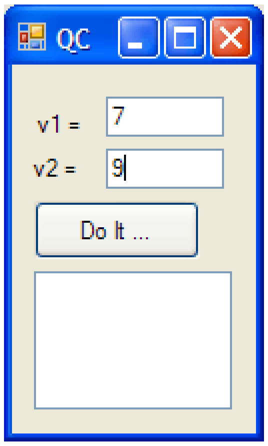
____

____

____

Notes

9. (2 points) Write down what is displayed in the listbox after the btnDoIt_Click sub in the program shown executes.



```
Public Class QC_9
    Private x As Double, indx As Integer
    Private Sub aSub(ByRef x As Double, _
                     ByVal y As Double)
        x = x + y
        indx = ListBox1.Items.Add("x = " & CStr(x))
    End Sub
    Private Sub btnDoIt_Click( ... ) _
                              Handles btnDoIt.Click
        Dim v1, v2 As Double
        v1 = Convert.ToDouble(txtV1.Text)
        v2 = Convert.ToDouble(txtV2.Text)
        x = 2
        Call aSub(v1, v2)
        indx = ListBox1.Items.Add("v1 = " & CStr(v1))
        indx = ListBox1.Items.Add("x = " & CStr(x))
    End Sub
End Class
```

Answer: ____________________

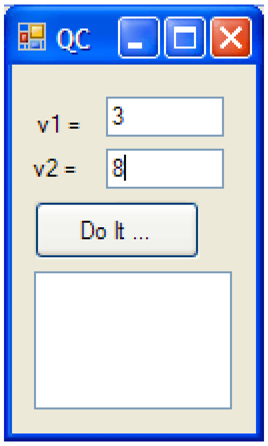
____

____

____

Notes

10. (2 points) Write down what is displayed in the listbox after the btnDoIt_Click sub in the program shown executes.



```
Public Class QC_10
    Private b As Double, indx As Integer
    Private Sub aSub(ByRef x As Double, _
                     ByVal y As Double)
        Dim b As Double
        b = x - y
        x = x + y
        indx = ListBox1.Items.Add("b = " & CStr(b))
    End Sub
    Private Sub btnDoIt_Click( ... ) _
                            Handles btnDoIt.Click
        Dim v1, v2 As Double
        v1 = Convert.ToDouble(txtV1.Text)
        v2 = Convert.ToDouble(txtV2.Text)
        b = 1
        Call aSub(v1, v2)
        indx = ListBox1.Items.Add("v1 = " & CStr(v1))
        indx = ListBox1.Items.Add("b = " & CStr(b))
    End Sub
End Class
```

Answer: ____________

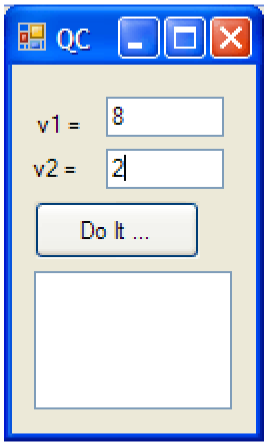
____

____

____

Notes

11. (2 points) Write down what is displayed in the listbox after the btnDoIt_Click sub in the program shown executes.



```
Public Class QC_11
    Private b As Double, indx As Integer
    Private Sub aSub(ByRef x As Double, _
                     ByVal y As Double)
        b = x - y
        indx = ListBox1.Items.Add("b = " & CStr(b))
    End Sub
    Private Sub btnDoIt_Click( ... ) _
                            Handles btnDoIt.Click
        Dim v1, v2 As Double
        v1 = Convert.ToDouble(txtV1.Text)
        v2 = Convert.ToDouble(txtV2.Text)
        b = 4
        indx = ListBox1.Items.Add("b = " & CStr(b))
        Call aSub(v1, v2)
        indx = ListBox1.Items.Add("b = " & CStr(b))
    End Sub
End Class
```

Answer: ____________

____

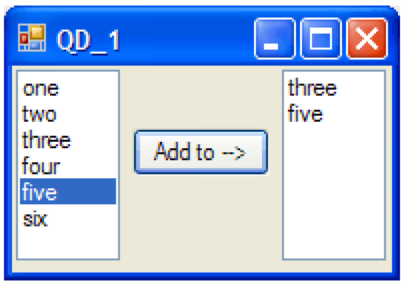
____

____

Notes

Question D (12 points)

1. (4 points) The form shown allows the user to select an item in the left-hand listbox and add it to the right-hand listbox by clicking the button. The user has already selected `three` and pressed the button, then `five` and pressed the button.



   The code for the click event of the Add to --> button requires just a variable declaration and one statement and uses the SelectedItem property of the listbox described in the object browser as:

   Public Property **SelectedItem**() As Object
   Member of: System.Windows.Forms.ListBox
   Summary: Gets or sets the currently selected item in the System.Windows.Forms.ListBox.
   Return Values: An object that represents the current selection in the control.

   Write the click event. The names for the listboxes are simply Listbox1 (the one on the left in the image) and Listbox2.

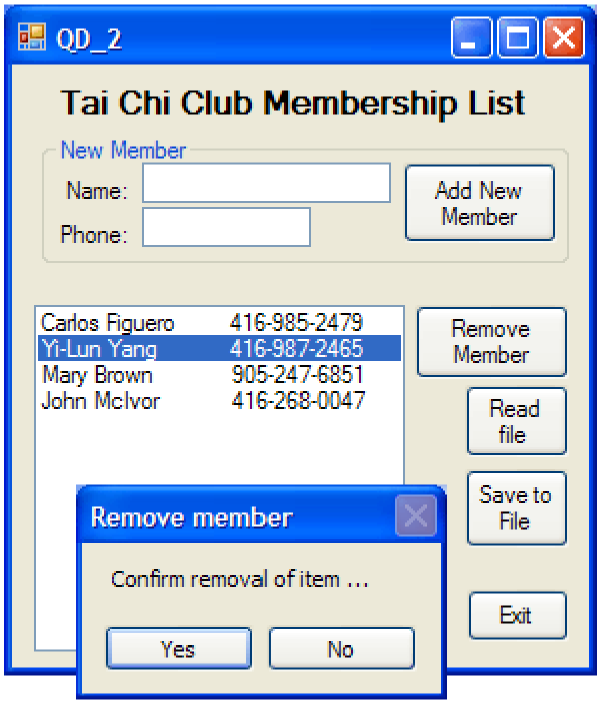
   Answer:

```
 Private Sub btnAddTo( ... ) Handles btnAddTo.Click




 End Sub
```

2. (8 points) The image shows an application used to maintain a membership list. Members are shown in the listbox (named lstMembers). Although there are other buttons we'll focus only on the one that removes a member from the list, which you will write.

   When the Remove Member button is clicked if there is an item selected the message box shown should be displayed, and only if the user clicks the Yes button should the item actually be removed. If no item is selected a different message box (having just an OK button), telling the user to first select a member should be displayed.

   Here is information from the object Browser about the Listbox, ObjectCollection and MessageBox classes that you'll need to use.



Carlos Figuero 416-985-2479
Yi-Lun Yang 416-987-2465
Mary Brown 905-247-6851
John McIvor 416-268-0047

Confirm removal of item ...

<u>From Listbox class:</u>

Public ReadOnly Property **Items**() As System.Windows.Forms.ListBox.ObjectCollection
Summary: Gets the items of the System.Windows.Forms.ListBox.
Return Values: An ListBox.ObjectCollection representing the items in the ListBox.

Public Property **SelectedIndex**() As Integer
Summary: Gets or sets the zero-based index of the currently selected item in a ListBox.
Return Values: A zero-based index of the currently selected item. A value of negative one (-1) is returned if no item is selected.

Public Property **SelectedItem**() As Object
Summary: Gets or sets the currently selected item in the ListBox.
Return Values: An object that represents the current selection in the control.

<u>From ObjectCollection class:</u>

Public Sub **Remove**(ByVal *value* As Object)
Summary: Removes the specified object from the collection.
Parameters: *value*: An object representing the item to remove from the collection.

<u>From MessageBox class:</u>

Public Shared Function **Show**(ByVal *text* As String**,** ByVal *caption* As String**,** ByVal *buttons* As System.Windows.Forms.MessageBoxButtons) As System.Windows.Forms.DialogResult
Summary: Displays a message box with specified text, caption, and buttons.
Parameters:
caption: The text to display in the title bar of the message box.
buttons: One of the System.Windows.Forms.MessageBoxButtons values that specifies which buttons to display in the message box.
text: The text to display in the message box.
Return Values: One of the System.Windows.Forms.DialogResult values.

Possible MessageBox buttons are: MessageBoxButtons.AbortRetryIgnore, MessageBoxButtons.OK, MessageBoxButtons.OKCancel, MessageBoxButtons.RetryCancel, MessageBoxButtons.YesNo, MessageBoxButtons.YesNoCancel

Possible DialogResult values are: DialogResult.Abort, DialogResult.Cancel, DialogResult.Ignore, DialogResult.No, DialogResult.OK, DialogResult.Retry, DialogResult.Yes

Write the btnRemove_Click sub.

```
Private Sub btnRemove_Click( ... ) Handles btnRemove.Click





End Sub
```

Question E (9 points)

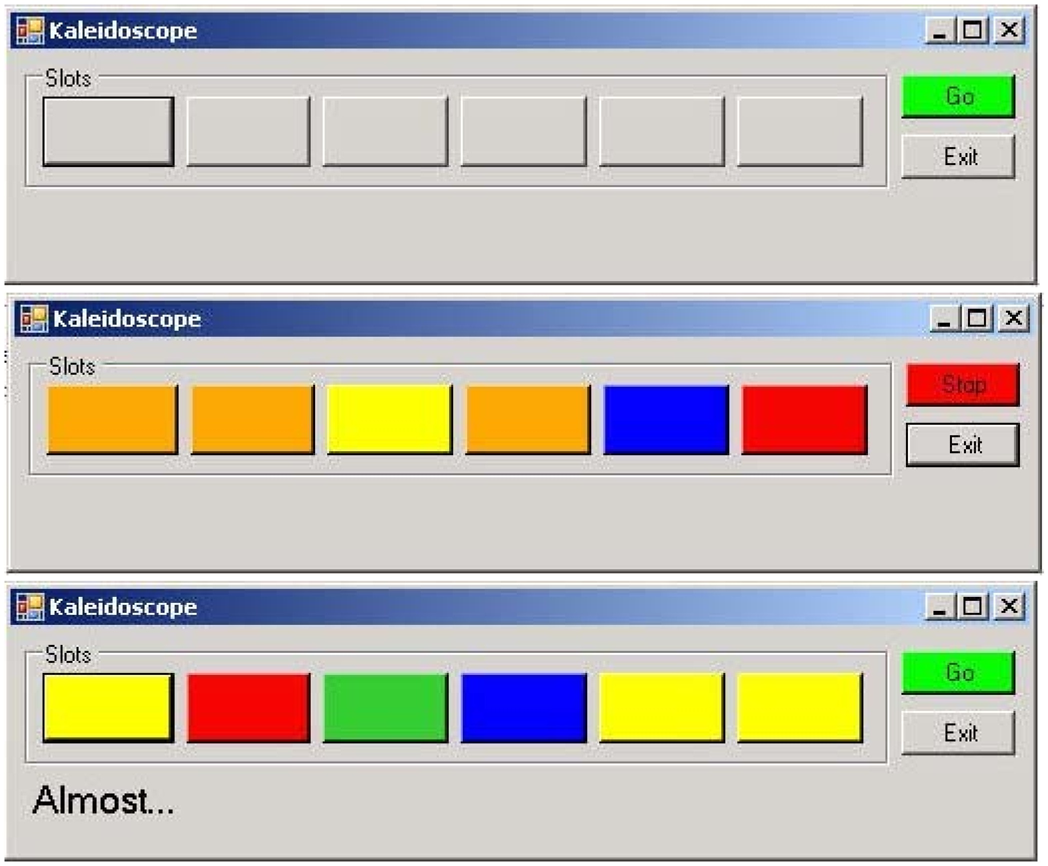
Be concise; more than a short sentence or two is not necessary!

1. (2 points) What general criteria should you use to choose names for variables, subprograms and control objects on the form?

2. (3 points) Explain, with reference to what occurs in the computer's main memory, the difference between a formal argument that is declared ByVal as opposed to ByRef.

3. (4 points) Give two reasons why subprograms are used when creating software.

Question F (20 points)
The 4 pages after this one contain the code that accompanies the Form displayed below, which implements a game of chance. The Go button causes the colours of the six buttons to continuously change randomly, and the user then clicks the Stop button while the colours are changing with the aim to stop them when as many colours as possible are the same. The more boxes that have the same colour, the bigger the “prize”.



Almost...

The Player begins by Clicking the Go button, which causes the following:

- the Go button is replaced by the Stop button (these two buttons are superimposed on each other)
- the six Slots begin flashing different colours at random

When the Player clicks Stop:

- the colours stop changing
- The Go button replaces the Stop button
- the number of slots of each colour are counted and a message is displayed depending on how many slots are the same colour

The interface consists of 9 buttons, 6 of which are in the gbxSlots groupbox. The other two, the Stop and Go buttons are superimposed on each other. There is also a label to display a message, named lblReward.

A `Timer` has also been included, but you don’t need to worry about how it works, only that it needs to be Enabled and Disabled.

Your task is to complete the code to make the game work. The lines that require attention are indicated by an arrow (➔).

```
Public Class Form1

  Private generateRandom As Random
 ________________________________________________________________________________
  Public Sub New()
    ' This call is required by the Windows Form Designer.
    InitializeComponent()
    ' Add any initialization after the InitializeComponent() call.
    generateRandom = New Random
  End Sub
 ________________________________________________________________________________
  Private Sub btnExit_Click(ByVal sender As System.Object, _
            ByVal e As System.EventArgs) Handles btnExit.Click
    Me.Close()
    Me.Dispose()
    End
  End Sub
 ________________________________________________________________________________
  Private Sub btnGo_Click(ByVal sender As System.Object, _
            ByVal e As System.EventArgs) Handles btnGo.Click
    '
    ' Hide/Show the appropriate controls, initialise the Reward label
    ' and enable the timer
    '
➔   __________________________________________

➔   __________________________________________

➔   __________________________________________

    Timer.Enabled = True
  End Sub
 ________________________________________________________________________________
  Private Sub Timer_Tick(ByVal sender As System.Object, _
            ByVal e As System.EventArgs) Handles Timer.Tick
    ' When the Timer is Enabled this code is executed 10 times/sec
    Dim colour As Color
    Dim ctrl As Byte
    ' loop through the controls in the Group
    For ctrl = 0 To grpSlots.Controls.Count - 1
        ' select a colour at random
        Call pick(colour)
        ' assign it to the control
        grpSlots.Controls(ctrl).BackColor = colour
        Next
  End Sub
 ________________________________________________________________________________
  '
  ' Complete the first and last line of the Sub.
  '
➔ Private _____________ pick (____________________________________________)
    '
    ' Called in Sub Timer_Tick.
    ' Select a random colour by generating a random integer in the
    ' range 1 - 6 inclusive; 1 = red, 2 = orange, 3 = yellow, etc.
    '
    Dim temp As Integer
    '
    ' Generate a random integer in the range 1 - 6 inclusive.
    '
    ' Also write this statement

➔   _______________________________________________________
    '
    ' Select the face colour according to its position in the rainbow (ROYGBV)
    '
        If temp = 1 Then
            face = Color.Red
        ElseIf temp = 2 Then
            face = Color.Orange
        ElseIf temp = 3 Then
            face = Color.Yellow
        ElseIf temp = 4 Then
            face = Color.LimeGreen
        ElseIf temp = 5 Then
            face = Color.Blue
        Else
            face = Color.Violet
        End If

➔  End_________________
________________________________________________________________________________
  Private Sub btnStop_Click(ByVal sender As System.Object, _
            ByVal e As System.EventArgs) Handles btnStop.Click
    '
    ' Disable the timer, hide/show the appropriate controls,
    ' call checkStatus sub to determine prize.
    '
    Timer.Enabled = False

➔  ______________________________

➔  ______________________________

    Call checkStatus()

  End Sub
________________________________________________________________________________
  '
  ' Complete the first and last line of the Sub.
  '
➔ Private  ________________ checkStatus _____________________
    '
    ' Also:
    ' build a string of letters representing the colours of the slots,
    ' find the appropriate message and display it
    '
    Dim colourString As String

➔   ____________________________________________
➔      _________________________________________

➔      _________________________________________

➔    End ____________________
```

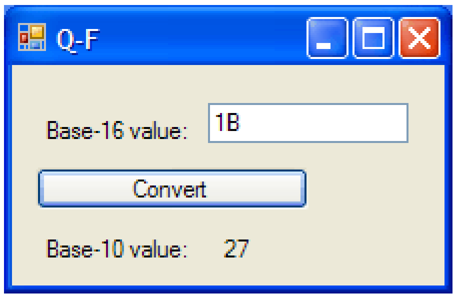
```
Private Sub build(ByRef colours As String)
  '
  ' Create a string representing the colours of the buttons by iterating
  ' through the controls in the groupbox nbamed grpSlots.
  '
   Dim ctrl As Byte
   For ctrl = 0 To grpSlots.Controls.Count - 1
       '
       ' Call a subprogram to select a letter representing the background
       ' colour of the button and add the letter to the colours string
       '
       ' Write the statement:

➔          ________________________________________________________
      Next
End Sub
```

```
  '
  ' Complete the first and last line of the Sub.
  '
➔ Private ___________________ colourOf________________________________________
    '
    ' The background colour of the sender control determines
    ' which letter to Return – called by the build Sub
    '
      If sender.BackColor = Color.Red Then
          Return "R"
      ElseIf sender.BackColor = Color.Orange Then
          Return "O"
      ElseIf sender.BackColor = Color.Yellow Then
          Return "Y"
      ElseIf sender.BackColor = Color.LimeGreen Then
          Return "G"
      ElseIf sender.BackColor = Color.Blue Then
          Return "B"
      Else
          Return "V"
      End If

➔ End ________________________
```

```
  '
  ' Complete the first and last line of the Sub.
  '
➔ Private ________________ reportOn _________________________________________
    '
    ' set the appropriate message for the number of like coloured buttons
    Dim m As String
```

```
        If count(colours) < 3 Then
            m = "Try again..."
        ElseIf count(colours) = 3 Then
            m = "Almost..."
        ElseIf count(colours) = 4 Then
            m = "4 of a kind pays 2-1"
        ElseIf count(colours) = 5 Then
            m = "5 of a kind pays 5-1"
        Else
             m = "** JACKPOT **"
         End If
         Return m

➔     End ____________________
    ________________________________________________________________________________
     '
     ' Complete the first and last line of the Sub.
     '
➔     Private _______________________ count____________________________________
        '
        ' count the number of slots of each colour
        '
        Dim p, R, O, Y, G, B, V As Byte
        Dim c As Char
        For p = 1 To Len(s)
            c = Mid(s, p, 1)
            If c = "R" Then
            R = R + 1
          ElseIf c = "O" Then
            O = O + 1
          ElseIf c = "Y" Then
            Y = Y + 1
          ElseIf c = "G" Then
            G = G + 1
          ElseIf c = "B" Then
            B = B + 1
          Else
            V = V + 1
          End If
        Next
        Return Math.Max(Math.Max(Math.Max(Math.Max(Math.Max(R, O), Y), G), B), V)

➔     End ________________________
    End Class
```

Question G (10 points)



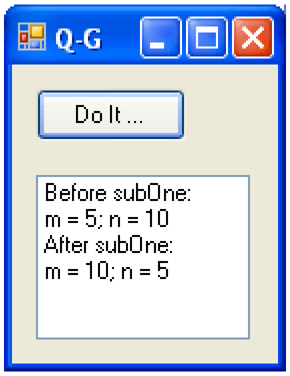
Exercise 5-11 of Visual Basic: Programming for Literacy requires you to write a program that converts a base-10 (i.e. decimal) integer to binary and vice versa. In this question you'll write a program (the click event for the Convert button) that converts a base-l6 integer to its base-l0 equivalent. A base-l6 integer might be `4A27E`, for example, which represents `4x16`^{4} `+ 10x16`^{3} `+2x16`^{2} `+ 7x16`^{1} `+15 x16`^{0}. Note that the "digit" A represents the tenth digit, B represents the eleventh, C represents the twelfth, D the thirteen, E the fourteenth, and F the fifteenth. (Any symbols could be used for these digits, but ABCDE and F are the convention.)

You may assume that the input is a valid base-16 value, i.e. it consists only of the characters 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, or F. Generally the first character(s) should not be 0 but this should not affect your calculation even if they were. By correctly using the constant DIGITS you can avoid writing an If statement. Declare other variables as necessary and use any sensible names for the control objects.

```
Public Class Form1

    Private Sub btnConvert_Click( ... ) Handles btnConvert.Click
        Const DIGITS = "0123456789ABCDEF"
```

```
    End Sub
End Class
```

Question H (7 points)



```
Before subOne:
m = 5; n = 10
After subOne:
m = 10; n = 5
```

1. (5 points) The image shows the form after the Do It … button has been pressed. The click event code is shown below. The click event calls a Sub named subOne. Your task is to write this sub. Notice that the values of the arguments **m** and **n** when the sub is called are 5 and 10, and that after the sub executes the values are 10 and 5, i.e. the values have been switched. I.e. the purpose of subOne is to switch the values of its arguments.

```
Private Sub Button1_Click( ... ) Handles Button1.Click
    Dim m, n As Integer
    Dim indx As Integer
    m = 5
    n = 10
    indx = ListBox1.Items.Add("Before subOne:")
    indx = ListBox1.Items.Add("m = " & CStr(m) & "; n = " & CStr(n))
    Call subOne(m, n)
    indx = ListBox1.Items.Add("After subOne:")
    indx = ListBox1.Items.Add("m = " & CStr(m) & "; n = " & CStr(n))
End Sub
```

Answer:

2. (2 points) If you were to display (e.g. by adding a Label to the Form) the value of the variable indx just before the click event ended what would that value be?

Answer: ___________________