We ran into some problems with the test for numeric and range.
I need to experiment with this a little.
It appears to be a change in the way things are handled. We will see...
Public Class frmAirfare
    Inherits System.Windows.Forms.Form
    Dim priceArray() As Single = {{800.97, 987.98, 299.99},
        {745.75, 923.45, 235.97},
        {907.07, 1128.90, 490.90},
        {708.08, 950.75, 259.98}}
    Dim fromCity(3) As String
    Dim toCity(2) As String

    Windows Form Designer generated code.

    Private Sub btnGetPrice_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnGetPrice.Click
        Dim whsfg As String
        If IsNumeric(txtFrom.Text) And CInt(txtFrom.Text) < 4 Then
            fromPtr = CInt(txtFrom.Text)
        "fromPtr = CInt(txtFrom.Text)
        If IsNumeric(txtTo.Text) And CInt(txtTo.Text) < 3 Then
            toPtr = CInt(txtTo.Text)
        "toPtr = CInt(txtTo.Text)
        wPrice = priceArray(fromPtr, toPtr)
    "wPrice = priceArray(fromPtr, toPtr)
        txtAirfare.Text = whsfg
        Else
            txtAirfare.Text = "From City is valid, To City is not valid"
        End If
    Else
        If IsNumeric(txtTo.Text) And CInt(txtTo.Text) < 3 Then
            txtAirfare.Text = "From City is not valid, To City is valid"
        Else
            txtAirfare.Text = "Both From City and To City are invalid"
        End If
    End If
End Sub

Private Sub frmAirfare_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    fromCity(0) = "Boston, MA"
    fromCity(1) = "New York, NY"
    fromCity(2) = "Austin, TX"
    fromCity(3) = "Washington, DC"
    toCity(0) = "Istanbul"
    toCity(1) = "Tokyo"
    toCity(2) = "London"
    priceArray(0, 0) = 800.97
    priceArray(0, 1) = 987.98
    priceArray(0, 2) = 299.99
    priceArray(0, 3) = 708.08
    priceArray(1, 0) = 874.75
    priceArray(1, 1) = 923.45
    priceArray(1, 2) = 235.97
    priceArray(1, 3) = 950.75
    priceArray(2, 0) = 1128.90
    priceArray(2, 1) = 490.90
    priceArray(2, 2) = 1178.80
    priceArray(2, 3) = 978.87
    priceArray(3, 0) = 299.99
    priceArray(3, 1) = 259.99
    priceArray(3, 2) = 745.75
    priceArray(3, 3) = 745.75

Title: Mar 12-12:58 PM (4 of 17)
Public Class frmArray

    Inherits System.Windows.Forms.Form

    Dim budgetArray(5) As Decimal

    Windows Form Designer generated code

    Private Sub btnGetData_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnGetData.Click
        Dim i As Integer
        Dim wkInfo As String
        For i = 0 To 5
            budgetArray(i) = InputBox("Enter budget for Dept " & i & ", Dept Budget")
            wkInfo = "Department " & i & ": " & FormatCurrency(budgetArray(i))
            lstShow.Items.Add(wkInfo)
        Next
    End Sub

    Private Sub btnProcess_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnProcess.Click
        Dim wkDeptBudget As Decimal
        Dim wkTotDeptBudget As Decimal
        For Each wkDeptBudget In budgetArray
            wkTotDeptBudget = wkTotDeptBudget + wkDeptBudget
        Next
        txtTotal.Text = FormatCurrency(wkTotDeptBudget)
    End Sub

End Class
Coding an internal sort.
Logic for the Bubble Sort:

Assume that we are starting out with the numbers 5, 6, 2, 3, 4 in that order.

SUB1 will point to the first thing we want to compare and SUB2 will point to the second. END-PT is set to how many numbers we are checking - since we are starting out at 5 numbers END-PT is initially set to 5.

FLIP-CT is incremented every time we flip - if we go through an entire pass without flipping we can assume the sort to be done. Otherwise the sort is done when we have done 4 passes (because there are 5 numbers to be sorted).

Pass 1:

<table>
<thead>
<tr>
<th>BEFORE</th>
<th>PROCESSING</th>
<th>SUB1</th>
<th>SUB2</th>
<th>END-PT</th>
<th>FLIP-CT</th>
<th>AFTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Initialize</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>Compare:</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>SUB1 pts at 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>SUB2 pts at 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>5&lt;6 leave alone</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>Compare:</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>SUB1 pts at 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>SUB2 pts at 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>6 not &lt; 2 so flip</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
</tr>
</tbody>
</table>
Private Sub btnSort_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSort.Click
    Dim i As Integer
    Dim endPt As Integer = 4
    Dim flipCnt As Integer = 0
    Dim sub1 As Integer = 9
    Dim sub2 As Integer = 9
    Dim holdSlot As String
    Do Until endPt = 0 Or flipCnt = 0
        sub1 = 0
        sub2 = 0
        flipCnt = 0
        Do Until sub2 > endPt
            If numArray(sub1) > numArray(sub2) Then
                holdSlot = numArray(sub1)
                numArray(sub1) = numArray(sub2)
                numArray(sub2) = holdSlot
                flipCnt = flipCnt + 1
            End If
            sub1 = sub1 + 1
            sub2 = sub2 + 1
        Loop
        endPt = endPt - 1
    Loop
    For i = 0 To 4
        lstSort.Items.Add(numArray(i))
    Next
End Sub
End Class
I am going to assign this topdown sort to be written by you in the way I did the bubble sort.

## Logic of Top-Down Sort:

15 These are the numbers I want to sort.

<table>
<thead>
<tr>
<th>Before</th>
<th>Processing</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Pass 1:**

<table>
<thead>
<tr>
<th>Before</th>
<th>Processing</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>Set up: Establish 2 subscripts: SUB1 = 1 and SUB2 = 2</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>Compare: 15 - what SUB1 is pointing to</td>
<td>15</td>
</tr>
<tr>
<td>24</td>
<td>36 - what SUB2 is pointing to</td>
<td>36</td>
</tr>
<tr>
<td>12</td>
<td>15 &lt; 36 so leave alone</td>
<td>24</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Before</th>
<th>Processing</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>Add 1 to SUB2 so, SUB2 = 3</td>
<td>15</td>
</tr>
<tr>
<td>36</td>
<td>Compare: 15 - what SUB1 is pointing to</td>
<td>36</td>
</tr>
<tr>
<td>24</td>
<td>24 - what SUB2 is pointing to</td>
<td>24</td>
</tr>
<tr>
<td>12</td>
<td>15 &lt; 24 so leave alone</td>
<td>24</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Before</th>
<th>Processing</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>Add 1 to SUB2 so, SUB2 = 4</td>
<td>15</td>
</tr>
<tr>
<td>36</td>
<td>Compare: 15 - SUB1 is pointed to</td>
<td>36</td>
</tr>
</tbody>
</table>
## Notes for CIS56 - Visual Basic

<table>
<thead>
<tr>
<th>Notes</th>
<th>Click on links to retrieve considerations.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCC lab handout</td>
<td>BCC Lab instructions to log in and use software.</td>
</tr>
<tr>
<td>Notes on DreamSpark</td>
<td>DreamSpark Support Register DreamSpark Account.</td>
</tr>
<tr>
<td></td>
<td>Download Access. The same techniques apply to other things you need to download.</td>
</tr>
<tr>
<td></td>
<td>Burn ISO to CD. Install and Register (Access).</td>
</tr>
<tr>
<td>Notes on ADO and Access</td>
<td>Notes on ADO Data Controls.</td>
</tr>
</tbody>
</table>

### Notes on Access

- Many examples taken from CIS126:
  - Sample Access 2010 database: Note there is an accompanying Presentation explaining this database under presentations (scroll down to see).
  - Note that the Access 2007 databases can be accessed using Access 2010.
  - In class project database using Access 2007.
  - Access Database for asg1.
  - Download steps for downloading Access Database.
  - In class project database using Access 2007.

**Presentations to accompany examples of Access 2010/2007:**

- Access 2010 example explanation - database in under examples.
- Zipped version of Access 2010 presentation and the database.
- Zipped version of Access 2007 introduction.
- Access 2007 inventory example.
- Presentation for asg1 - Assignment #1.
- Separate speaker notes to accompany asg1.
- Assignment #1 for Access 2007 in pdf format.
- Presentation for if queries in Access.
Top down sort presentation.
Sorting arrays/tables
Top Down sort

Please use speaker notes for additional information!
The sort of the array provided by Microsoft.
Public Class frmSortMethod

    Private Dim numArray() As Integer

    Private Sub btnEnter_Click(ByVal sender As Object, ByVal e As System.EventArgs)
        Dim i As Integer
        For i = 0 To 4
            numArray(i) = InputBox("Enter Number", "Sort")
            lstEnter.Items.Add(numArray(i))
        Next
    End Sub

    Private Sub btnSort_Click(ByVal sender As Object, ByVal e As System.EventArgs)
        Array.Sort(numArray)
        For i = 0 To 4
            lstSort.Items.Add(numArray(i))
        Next
    End Sub

End Class
Sort and display in reverse order.