Public Class frmArray
    Inherits System.Windows.Forms.Form
    Dim budgetArray(15) As Decimal

    Windows Form Designer generated code

    Private Sub btnGetData_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Dim i As Integer
        Dim wkInfo As String
        Dim wkMax As Integer
        wkMax = InputBox("Enter the number of departments being processed", "Department Count")
        budgetArray(0) = 0
        For i = 1 To wkMax
            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)
        Dim wkTotBudget As Decimal = 0
        Dim wkTotCt As Integer = 0
        Dim i As Integer
        For i = 1 To budgetArray.Length - 1
            wkTotBudget = wkTotBudget + budgetArray(i)
            If Not budgetArray(i) = 0 Then
                wkTotCt = wkTotCt + 1
            End If
        Next
        txtTotal.Text = FormatCurrency(wkTotBudget)
        txtDeptNum.Text = wkTotCt
    End Sub

End Class
Public Class frmArray

    Inherits System.Windows.Forms.Form

    Dim budgetArray(15) 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 wklInfo As String
        Dim wkMax As Integer
        wkMax = InputBox("Enter the number of departments below")
        budgetArray(0) = 0
        For i = 1 To wkMax
            budgetArray(i) = InputBox("Enter budget")
            wklInfo = "Department ", i & ", ", For Each i = lstShow.Items.Add(wklInfo)
            Next
        End Sub

    Private Sub btnProcess_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnProcess.Click
        Dim wkTotBudget As Decimal = 0
        Dim wkTotCt As Integer = 0
        Dim whatever As Integer
        whatever = budgetArray.Length
        For i = 1 To BudgetArray.Length - 1
            wkTotBudget = wkTotBudget + budgetArray(i)
            If Not budgetArray(i) = 0 Then
                wkTotCt = wkTotCt + 1
            End If
        Next
        txtTotal.Text = FormatCurrency(wkTotBudget)
        txtDeptNum.Text = wkTotCt
    End Sub

End Class
Alternative ways to test for not equal.
Inherits System.Windows.Forms.Form

Dim budgetArray(15) As Decimal

Windows Form Designer generated code

Private Sub btnGetData_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Dim i As Integer
    Dim wInfo As String
    Dim wMax As Integer
    wMax = InputBox("Enter the number of departments being processed", "Department Count")
    budgetArray(0) = 0
    For i = 1 To wMax
        budgetArray(i) = InputBox("Enter budget for Dept ", "Dept Budget")
        wInfo = "Department ", i + 1, ", " n FormatCurrency(budgetArray(i))
        lstShow.Items.Add(wInfo)
    Next
End Sub

Private Sub btnProcess_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Dim wTotBudget As Decimal = 0
    Dim wTotCt As Integer = 0
    Dim i As Integer
    Dim whatever As Integer
    whatever = budgetArray.Length
    MsgBox(whatever)
    For i = 1 To budgetArray.Length - 1
        wTotBudget = wTotBudget + budgetArray(i)
        If Not budgetArray(i) = 0 Then
            If budgetArray(i) <> 0 Then
                wTotCt = wTotCt + 1
            End If
        End If
    Next
    txTotal.Text = FormatCurrency(wTotBudget)
    txtPercent.Text = wTotCt
End Sub
Parallel arrays

```vbnet
Public Class frmDept
    Inherits System.Windows.Forms.Form
    Dim deptNumArray(3) As Integer
    Dim deptArray(3) As String

    Windows Form Designer generated code

    Private Sub frmDept_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        deptNumArray(0) = 15
        deptNumArray(1) = 26
        deptNumArray(2) = 37
        deptNumArray(3) = 56
        deptArray(0) = "Books"
        deptArray(1) = "Toys"
        deptArray(2) = "Gifts"
        deptArray(3) = "Cookware"
    End Sub

    Private Sub btnListDept_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnListDept.Click
        Dim i As Integer
        For i = 0 To 3
            listDept.Items.Add(deptNumArray(i) & " " & deptArray(i))
        Next
    End Sub
End Class
```
Public Class frmAirFare
Inherits System.Windows.Forms.Form
Dim fromCity(3) As String
Dim toCity(2) As String
Dim priceArray(5, 2) As Single

Private Sub btnGetPrice_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
  Dim fromPtr As Integer, toPtr As Integer, wkPrice As Single
  Dim wMsg As String
  If IsNumeric(txtFrom.Text) And CInt(txtFrom.Text) < 4 Then
    fromPtr = CInt(txtFrom.Text)
    If IsNumeric(txtTo.Text) And CInt(txtTo.Text) < 3 Then
      toPtr = CInt(txtTo.Text)
      wkPrice = priceArray(fromPtr, toPtr)
      wMsg = "The price to fly from " & fromCity(fromPtr) & " to " & toCity(toPtr) & " is " & Currency(wkPrice)
      txtFare.Text = wMsg
    Else
      txtFare.Text = "From City is valid, To City is not valid"
    End If
  Else
    txtFare.Text = "From City is not valid, To City is invalid"
  End If
End Sub

Private Sub frmAirFare_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
  fromCity(0) = "Tokyo, Jp"
End Sub

Two dimensional array.
Private Sub frmAirfare_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Dim fromCity(0) = New String() {"Boston, MA", "New York, NY", "Austin, TX", "Washington, DC"}
    Dim toCity(0) = New String() {"Istanbul", "Beijing", "London", "Tokyo"}
    Dim priceArray(0, 0) As Double = New Double() {
        {600.97, 587.96, 923.45, 235.97, 967.67, 1120.98, 498.96, 750.75, 980.75, 230.98}
        {587.96, 923.45, 235.97, 967.67, 1120.98, 498.96, 750.75, 980.75, 230.98}
        {923.45, 235.97, 967.67, 1120.98, 498.96, 750.75, 980.75, 230.98}
        {235.97, 967.67, 1120.98, 498.96, 750.75, 980.75, 230.98}
    }
    ' Your code here...
End Sub

Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCancel.Click
    txtFrom.Clear()
    txtTo.Clear()
    txtAirfare.Clear()
End Sub

Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSave.Click
    ' Your code here...
End Sub

Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnExit.Click
    Me.Close()
End Sub
Private Sub frmAirfare_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles sender.Load
    Dim fromCity As String
    Dim toCity As String
    Dim price As Double

    fromCity(0) = "Boston, MA"
    fromCity(1) = "New York, NY"
    fromCity(2) = "Austin, TX"
    fromCity(3) = "Washington, DC"
    toCity(0) = "Istanbul"
    toCity(1) = "Paris"
    toCity(2) = "London"

    Dim priceArray(3, 3) As Double
    priceArray(0, 0) = 600.99
    priceArray(0, 1) = 967.99
    priceArray(0, 2) = 599.99
    priceArray(0, 3) = 745.75
    priceArray(1, 0) = 923.45
    priceArray(1, 1) = 235.97
    priceArray(1, 2) = 967.89
    priceArray(1, 3) = 1120.00
    priceArray(2, 0) = 496.98
    priceArray(2, 1) = 750.75
    priceArray(2, 2) = 590.75
    priceArray(2, 3) = 250.98
    priceArray(3, 0) = 1100.99
    priceArray(3, 1) = 1300.00
    priceArray(3, 2) = 1500.00

    Dim fromCode As Integer
    Dim toCode As Integer
    Dim priceCost As Double

    From Code Box: 2
    To Code Box: 1

    priceCost = priceArray(fromCode - 1, toCode - 1)
    MessageBox.Show("The price to fly from " & fromCity(toCode - 1) & " to " & toCity(fromCode - 1) & " is: 
        " & priceCost.ToString("C0"))
End Sub
Public Class frmAirFace
    Inherits System.Windows.Forms.Form

    Dim fromCity(3) As String
    Dim toCity(2) As String
    Dim priceArray(3, 2) As Single

End Class

Private Sub btnGetPrice_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnGetPrice.Click
    Dim fromPtr As Integer, toPtr As Integer, wkPrice As Single
    Dim wkMsg As String

    If IsNumeric(txtFrom.Text) And CInt(txtFrom.Text) < 4 Then
        fromPtr = CInt(txtFrom.Text)
    ElseIf IsNumeric(txtTo.Text) And CInt(txtTo.Text) < 3 Then
        toPtr = CInt(txtTo.Text)
    End If

    wkPrice = priceArray(fromPtr, toPtr)
    wkMsg = String.Format("The price to fly from " & fromCity(fromPtr) & " to " & toCity(toPtr) & " is ", FormatCurrency(wkPrice))
    txtAirFace.Text = wkMsg

    Else
        txtAirFace.Text = "From City is valid, To City is not valid"
    End If

    ElseIf IsNumeric(txtFrom.Text) And CInt(txtTo.Text) < 3 Then
        txtAirFace.Text = "From City is not valid, To City is valid"
    ElseIf IsNumeric(txtTo.Text) And CInt(txtFrom.Text) < 4 Then
        txtAirFace.Text = "Both From City and To City are invalid"
    End If
End Sub

Private Sub frmAirFace_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles frmAirFace.Load
    fromCity(0) = "Boston, MA"
    toCity(0) = "New York, NY"
    toCity(1) = "Los Angeles, CA"
    toCity(2) = "Chicago, IL"
End Sub
Private Sub btnGetPrice_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnGetPrice.Click
    Dim fromPtr As Integer, toPtr As Integer, wkPrice As Single
    Dim Working As String
    If IsNumeric(txtFrom.Text) And CInt(txtFrom.Text) < 4 Then
        fromPtr = CInt(txtFrom.Text)
    ElseIf IsNumeric(txtTo.Text) And CInt(txtTo.Text) < 3 Then
        toPtr = CInt(txtTo.Text)
    End If
    wkPrice = priceArray(fromPtr, toPtr)
    Working = "The price to fly from " & fromCity(fromPtr) & " to " & toCity(toPtr) & " is " & FormatCurrency(wkPrice)
    txt fare.Text = Working
Else
    txt fare.Text = "From City is valid, To City is not valid"
End If
Else
    If IsNumeric(txtTo.Text) And CInt(txtTo.Text) < 3 Then
        txt fare.Text = "From City is not valid, To City is valid"
    ElseIf IsNumeric(txtFrom.Text) AND CInt(txtFrom.Text) < 4 Then
        txt fare.Text = "Both From City and To City are invalid"
    End If
End If
End Sub
Public Class frmAirFace
    Inherits System.Windows.Forms.Form
    Dim FromCity(3) As String
    Dim toCity(2) As String
    Dim priceArray(3, 2) As Single

    Windows Form Designer generated code

    Private Sub btnGetPrice_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Dim fromPtr As Integer, toPtr As Integer, wkPrice As Single
        Dim wMsg As String
        If IsNumeric(txtFrom.Text) And CInt(txtFrom.Text) < 4 Then
            fromPtr = CInt(txtFrom.Text)
        Else
            txtFace.Text = "From City is invalid, To City is not valid"
            End If
        End If
        If IsNumeric(txtTo.Text) And CInt(txtTo.Text) < 3 Then
            toPtr = CInt(txtTo.Text)
        Else
            txtFace.Text = "From City is not valid, To City is valid"
            End If
        End If
        If fromPtr = toPtr Then
            wMsg = "The price to fly from " & FromCity(fromPtr) & _
                " to " & toCity(toPtr) & _
                " is " & FormatCurrency(wkPrice)
            txtFace.Text = wMsg
        Else
            wMsg = "The price to fly from " & FromCity(fromPtr) & _
                " to " & toCity(toPtr) & _
                " is " & _
                FormatCurrency(wkPrice)
            txtFace.Text = wMsg
        End If
    End Sub

    Private Sub frmAirFace_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        FromCity(0) = "Pittsburgh, PA"
    End Sub
The PowerPoint presentation is under presentations.

Bubble sort - PASS #1

Before numbers:

5  2  3  6  4

SUB1

1  2  3  4

SUB2

2  3  4  5

HOLD 6

END-PT 5

FLIP-CT

0  1  2  3

The PowerPoint presentation is under presentations.

After numbers:

5  2  3  4  6

Compare 6 which is what SUB1 is pointing to and 4 which is what SUB2 is pointing to.

In this instance, $6 > 4$ so flip and add one to the flip-ct.

At this point we notice that 5 has sunk to the bottom.
Uses the sort method.
When you sort, you will probably use the sort provided by VB.
Public Class frmSortMethod
    Inherits System.Windows.Forms.Form
    Dim numArray(4) As Integer

    Windows Form Designer generated code

Private Sub btnEnter_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnEnter.Click
    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 System.Object, ByVal e As System.EventArgs) Handles btnSort.Click
    Dim i As Integer
    Array.Sort(numArray)
    For i = 0 To 4
        lstSort.Items.Add(numArray(i))
    Next
End Sub
End Class
Here the program uses its own sort algorithm, the bubble sort. The steps are illustrated in the bubble sort PowerPoint.
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
    Dim sub2 As Integer
    Dim holdSlot As String

    Do Until endPt = 0 Or flipCnt = 0
        sub1 = 0
        sub2 = 1
        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
        ListSort.Items.Add(numArray(i))
    Next

    End Sub
End Class
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
    Dim sub2 As Integer
    Dim holdSlot As Integer

    Do Until endPt = 0 Or flipCnt = 0
        sub1 = 0
        sub2 = 1
        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
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 flipCt As Integer = 0
    Dim sub1 As Integer
    Dim sub2 As Integer
    Dim holdSlot As Integer
    Do Until endPt = 0 Or flipCt = 0
        sub1 = 0
        sub2 = 1
        flipCt = 0
        Do Until sub2 > endPt
            If numArray(sub1) > numArray(sub2) Then
                holdSlot = numArray(sub1)
                numArray(sub1) = numArray(sub2)
                numArray(sub2) = holdSlot
                flipCt = flipCt + 1
            End If
            sub1 = sub1 + 1
            sub2 = sub2 + 1
        Loop
        endPt = endPt - 1
    Loop
    For i = 0 To 4
        ListSort.Items.Add(numArray(i))
    Next
End Sub
There is also a PowerPoint for the top down sort which I am going to ask you to write.
This shows the array defined as a structure that I have developed. I have done this in a module so it is public and available to both of my forms.
Under project, I can add forms and add modules.
The deptNo and deptTot were defined in my structure.
Public Class frmDataEntry
    Inherits System.Windows.Forms.Form

    Windows Form Designer generated code

    Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAdd.Click
        Dim wkSub As Integer = Integer.Parse(txtDeptNo.Text) - 1
        deptArray(wkSub).deptNo = deptArray(wkSub).deptNo + CDbl(txtAmt.Text)
    End Sub

    Private Sub frmDataEntry_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        deptArray(0).deptNo = 1
        deptArray(1).deptNo = 2
        deptArray(2).deptNo = 3
        deptArray(3).deptNo = 4
        deptArray(4).deptNo = 5
        deptArray(5).deptNo = 6
    End Sub

    Private Sub btnFinish_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnFinish.Click
        Dim sortShowForm As New frmSortedTotals
        sortShowForm.ShowDialog()
    End Sub
End Class
Private Sub btnSort_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSort.Click
    Dim endPt As Integer = 5
    Dim flipCnt As Integer = 9
    Dim holdDeptNo As Integer
    Dim holdDeptTot As Decimal
    Dim sub1 As Integer
    Dim sub2 As Integer
    Dim i As Integer
    Dim vLine As String

    Do Until endPt = 0 Or flipCnt = 0
        sub1 = 0
        sub2 = 1
        flipCnt = 0
        Do Until sub2 > endPt
            If deptArray(sub1).deptTot > deptArray(sub2).deptTot Then
                holdDeptNo = deptArray(sub1).deptNo
                holdDeptTot = deptArray(sub1).deptTot
                deptArray(sub1).deptNo = deptArray(sub2).deptNo
                deptArray(sub1).deptTot = deptArray(sub2).deptTot
                deptArray(sub2).deptNo = holdDeptNo
                deptArray(sub2).deptTot = holdDeptTot
                flipCnt = flipCnt + 1
        End If
        sub1 = sub1 + 1
        sub2 = sub2 + 1
    Loop
    endPt = endPt - 1
    Loop
    For i = 0 To 5
        vLine = "The total for Dept # " & deptArray(i).deptNo & ", " & FormatCurrency(deptArray(i).deptTot) & ".
        listTotals.Items.Add(vLine)
    Next
Public Class frmSortedTotals
    Inherits System.Windows.Forms.Form

    Windows Form Designer generated code

    Private Sub btnReturn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnReturn.Click
        Me.Close()
    End Sub

    Private Sub btnSort_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSort.Click
        Dim endPt As Integer = 5
        Dim flipCt As Integer = 9
        Dim holdDeptNo As Integer
        Dim holdDeptTot As Decimal
        Dim sub1 As Integer
        Dim sub2 As Integer
        Dim i As Integer
        Dim sLine As String
        Do Until endPt = 0 Or flipCt = 0
            sub1 = 0
            sub2 = 1
            flipCt = 0
            Do Until sub2 > endPt
                If deptArray(sub1).deptTot > deptArray(sub1).deptTot Then
                    holdDeptNo = deptArray(sub1).deptNo
                    holdDeptTot = deptArray(sub1).deptTot
                    deptArray(sub1).deptNo = deptArray(sub2).deptNo
                    deptArray(sub1).deptTot = deptArray(sub2).deptTot
                    deptArray(sub2).deptNo = holdDeptNo
                    deptArray(sub2).deptTot = holdDeptTot
                    flipCt = flipCt + 1
                End If
                sub1 = sub1 + 1
                sub2 = sub2 + 1
            Loop
        Loop
    End Sub