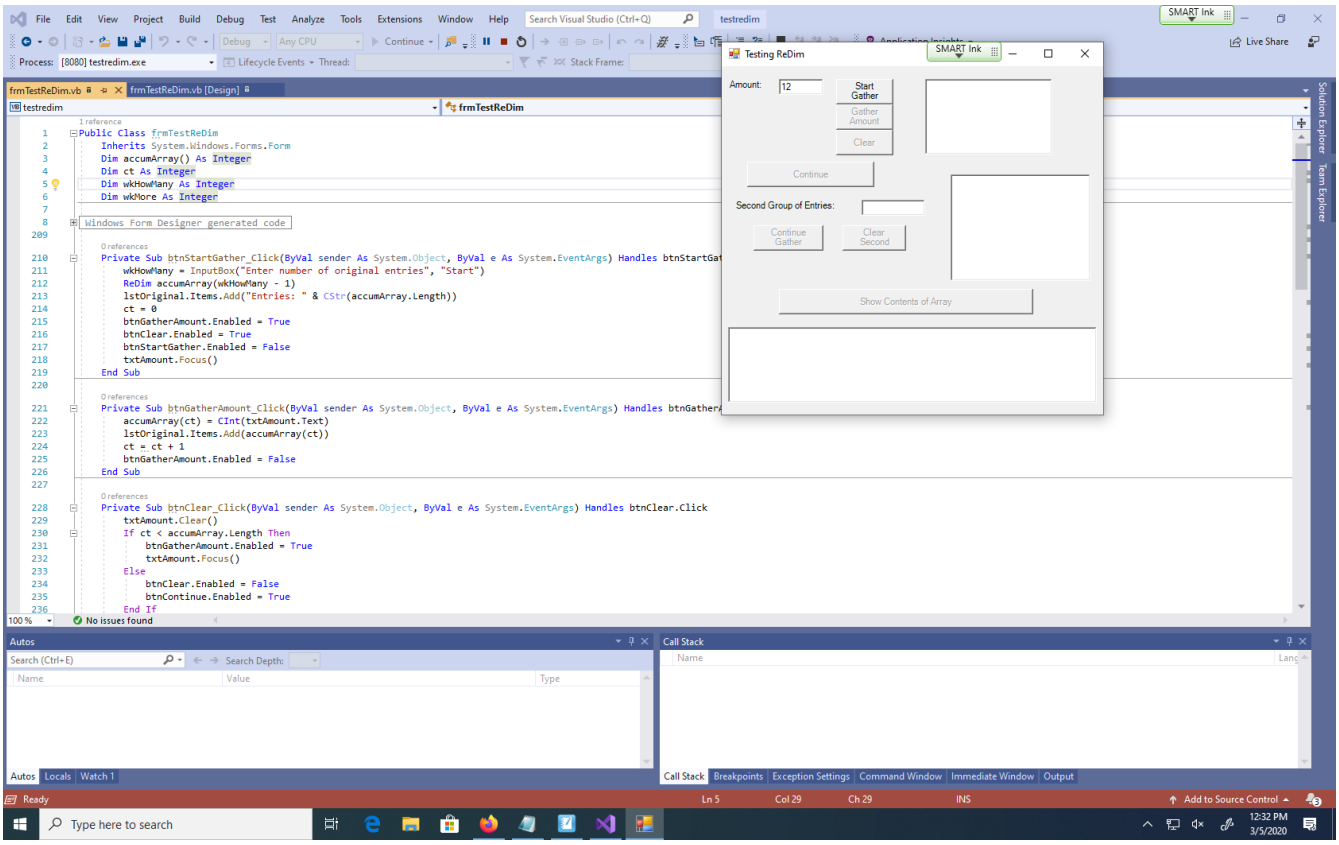
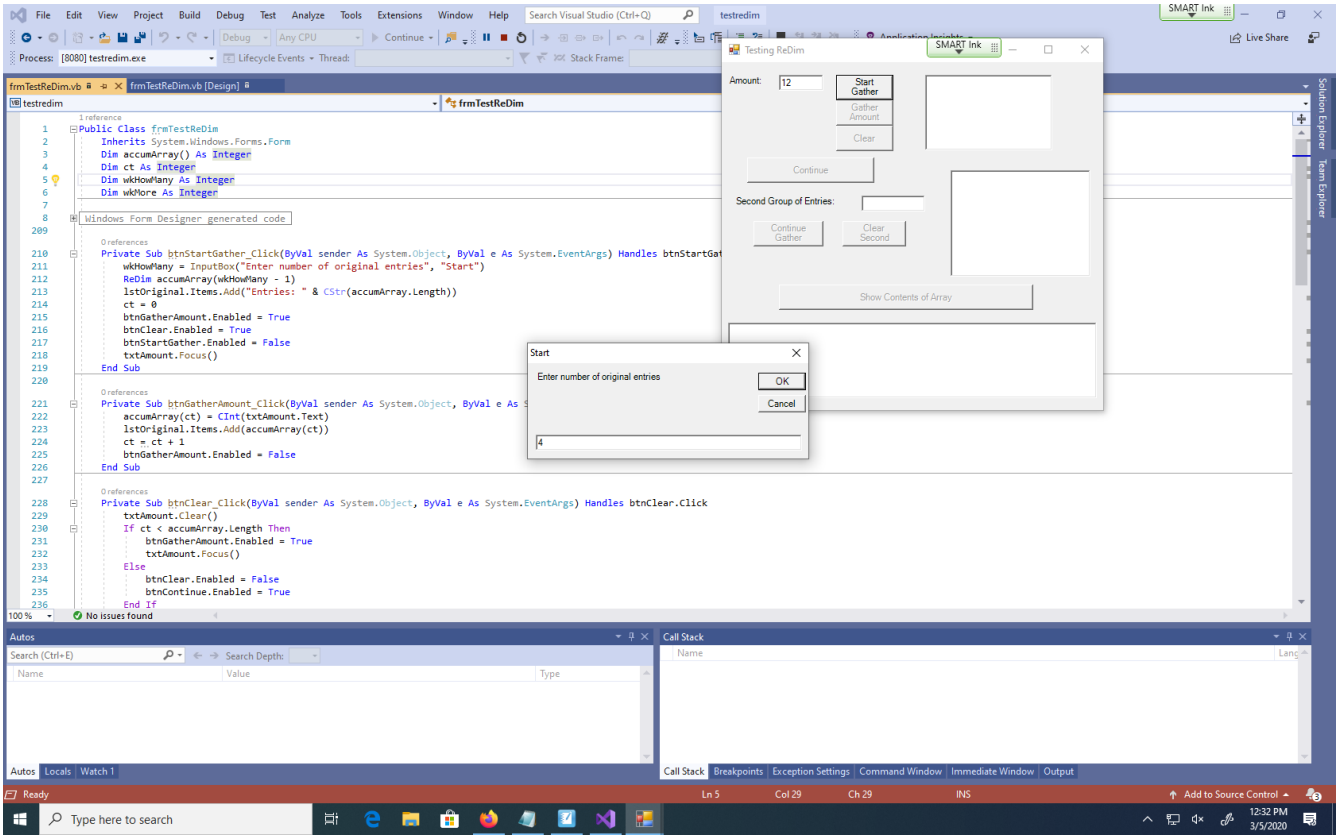
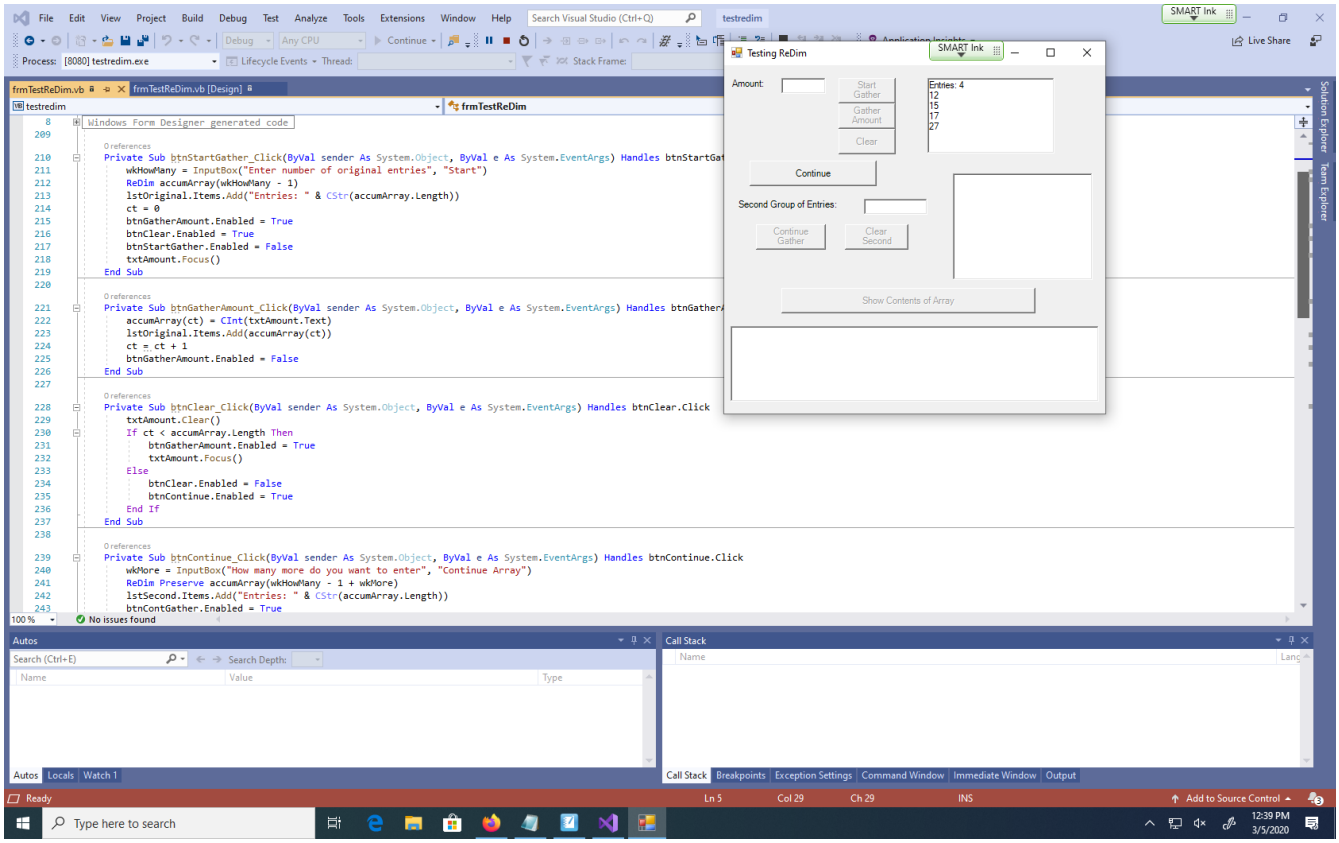
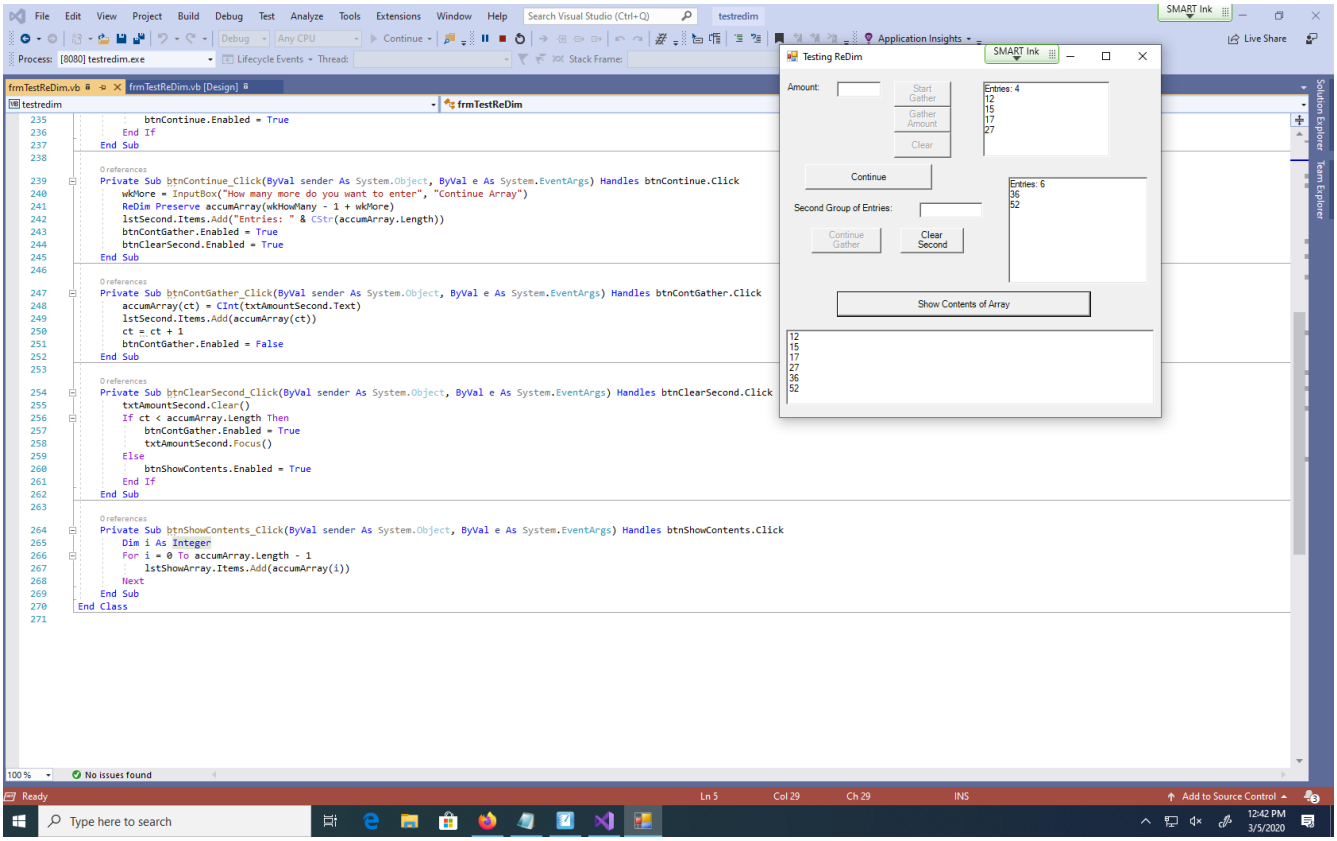
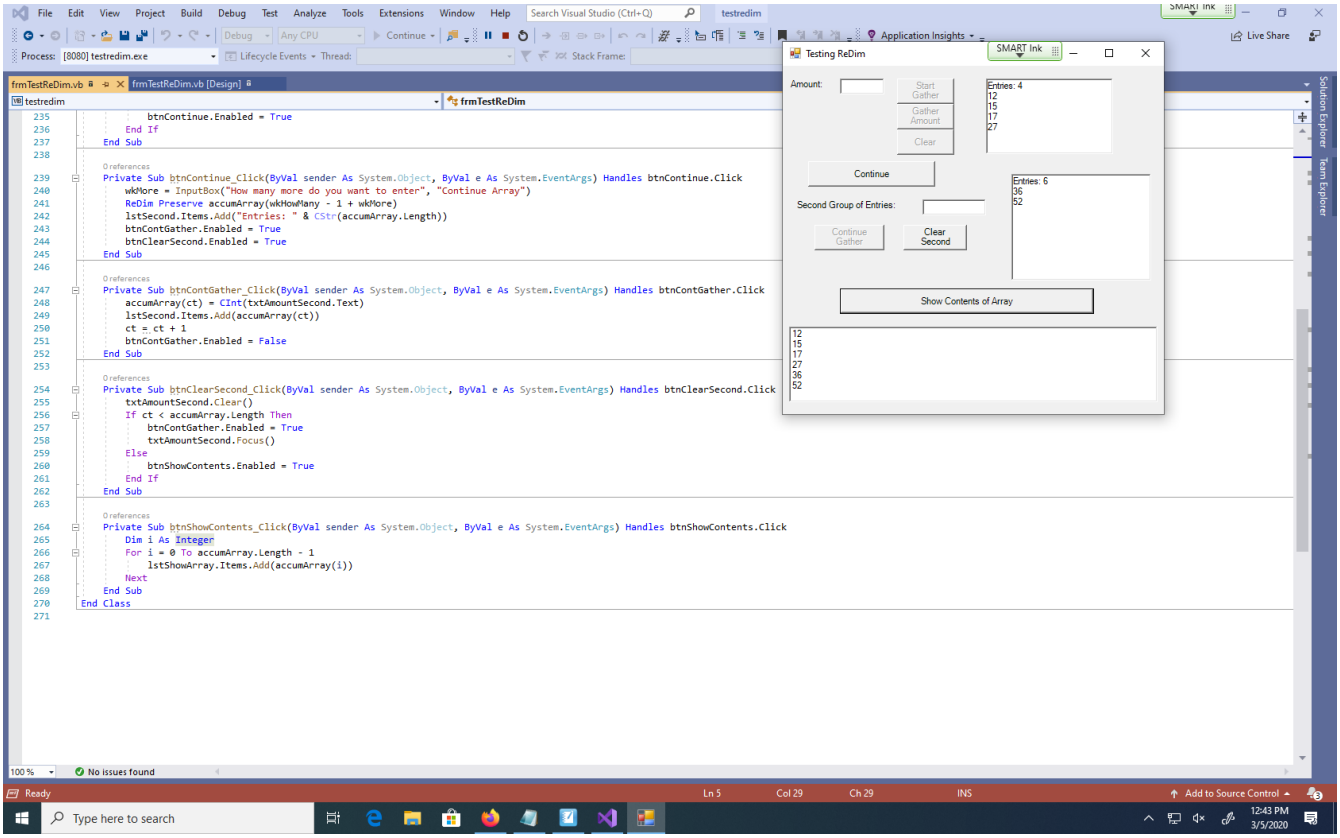
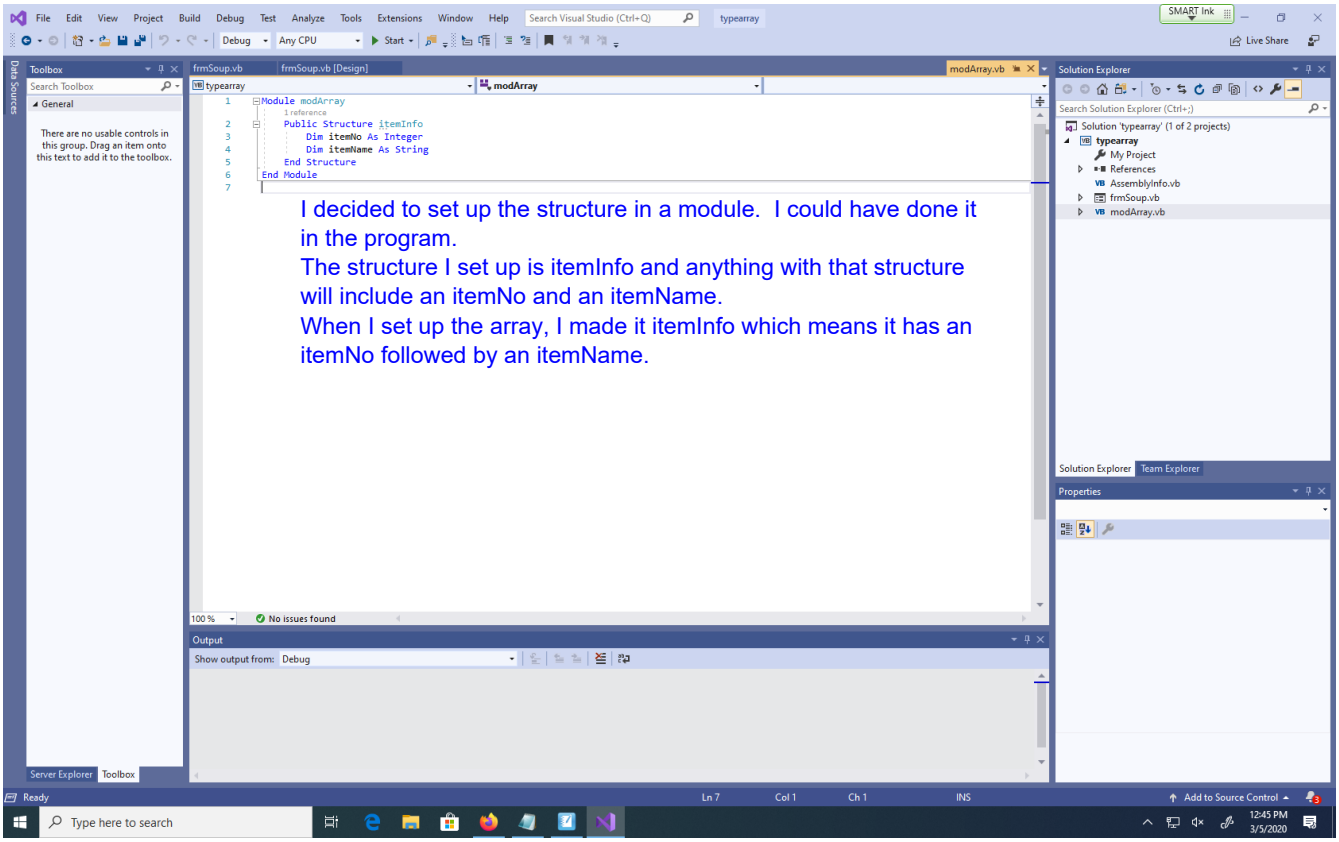

Testing ReDim
Amount:
12
Start Gather
Gather Amount
Clear
Continue
Second Group of Entries:
Continue Gather
Clear Second
Show Contents of Array
Public Class frmTestReDim
Inherits System.Windows.Forms.Form
Dim accumArray() As Integer
Dim ct As Integer
Dim wkHowMany As Integer
Dim wkMore As Integer
Windows Form Designer generated code
Private Sub btnStartGather_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnStartGat
wkHowMany = InputBox("Enter number of original entries", "Start")
ReDim accumArray(wkHowMany - 1)
lstOriginal.Items.Add("Entries: " & CStr(accumArray.Length))
ct = 0
btnGatherAmount.Enabled = True
btnClear.Enabled = True
btnStartGather.Enabled = False
txtAmount.Focus()
End Sub
Private Sub btnGatherAmount_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnGatherA
accumArray(ct) = CInt(txtAmount.Text)
lstOriginal.Items.Add(accumArray(ct))
ct = ct + 1
btnGatherAmount.Enabled = False
End Sub
Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClear.Click
txtAmount.Clear()
If ct < accumArray.Length Then
btnGatherAmount.Enabled = True
txtAmount.Focus()
Else
btnClear.Enabled = False
btnContinue.Enabled = True
End If

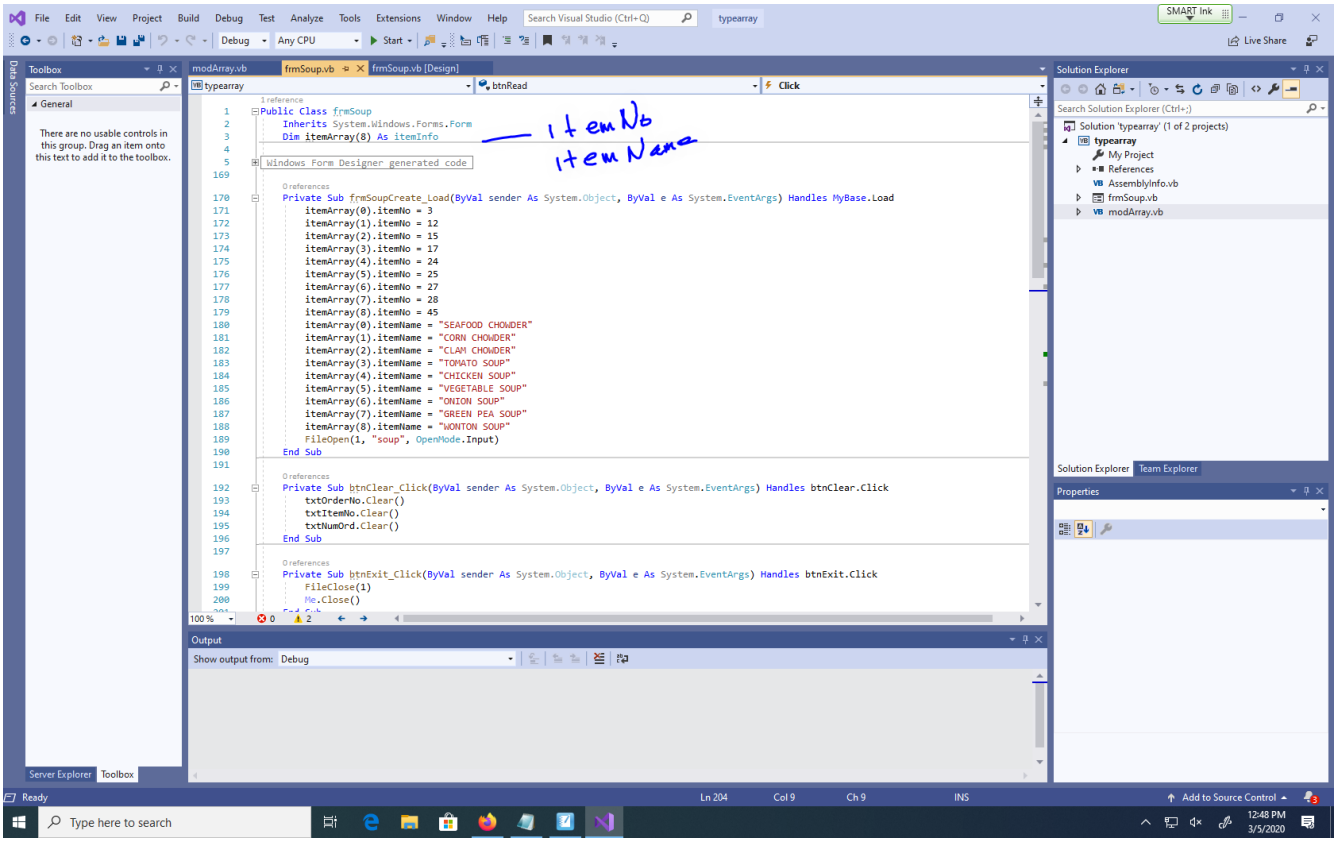

Testing ReDim
Amount:
12
Start Gather
Gather Amount
Clear
Continue
Second Group of Entries:
Continue Gather
Clear Second
Show Contents of Array
Start
Enter number of original entries
OK
Cancel
4
Public Class frmTestReDim
Inherits System.Windows.Forms.Form
Dim accumArray() As Integer
Dim ct As Integer
Dim wkHowMany As Integer
Dim wkMore As Integer
Windows Form Designer generated code
Private Sub btnStartGather_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnStartGat
wkHowMany = InputBox("Enter number of original entries", "Start")
ReDim accumArray(wkHowMany - 1)
lstOriginal.Items.Add("Entries: " & CStr(accumArray.Length))
ct = 0
btnGatherAmount.Enabled = True
btnClear.Enabled = True
btnStartGather.Enabled = False
txtAmount.Focus()
End Sub
Private Sub btnGatherAmount_Click(ByVal sender As System.Object, ByVal e As
accumArray(ct) = CInt(txtAmount.Text)
lstOriginal.Items.Add(accumArray(ct))
ct = ct + 1
btnGatherAmount.Enabled = False
End Sub
Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClear.Click
txtAmount.Clear()
If ct < accumArray.Length Then
btnGatherAmount.Enabled = True
txtAmount.Focus()
Else
btnClear.Enabled = False
btnContinue.Enabled = True
End If
No issues found
Autos
Call Stack
Ready
Type here to search


Testing ReDim
Amount:
Start Gather
Gather Amount
Clear
Entries: 4
12
15
17
27
Continue
Second Group of Entries:
Continue Gather
Clear Second
Show Contents of Array
frmTestReDim.vb
frmTestReDim.vb [Design]
testredim
frmTestReDim
Windows Form Designer generated code
Private Sub btnStartGather_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnStartGat
wkHowMany = InputBox("Enter number of original entries", "Start")
ReDim accumArray(wkHowMany - 1)
lstOriginal.Items.Add("Entries: " & CStr(accumArray.Length))
ct = 0
btnGatherAmount.Enabled = True
btnClear.Enabled = True
btnStartGather.Enabled = False
txtAmount.Focus()
End Sub
Private Sub btnGatherAmount_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnGather
accumArray(ct) = CInt(txtAmount.Text)
lstOriginal.Items.Add(accumArray(ct))
ct = ct + 1
btnGatherAmount.Enabled = False
End Sub
Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClear.Click
txtAmount.Clear()
If ct < accumArray.Length Then
btnGatherAmount.Enabled = True
txtAmount.Focus()
Else
btnClear.Enabled = False
btnContinue.Enabled = True
End If
End Sub
Private Sub btnContinue_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnContinue.Click
wkMore = InputBox("How many more do you want to enter", "Continue Array")
ReDim Preserve accumArray(wkHowMany - 1 + wkMore)
lstSecond.Items.Add("Entries: " & CStr(accumArray.Length))
btnContGather.Enabled = True
No issues found
Autos
Call Stack
Ready
Type here to search

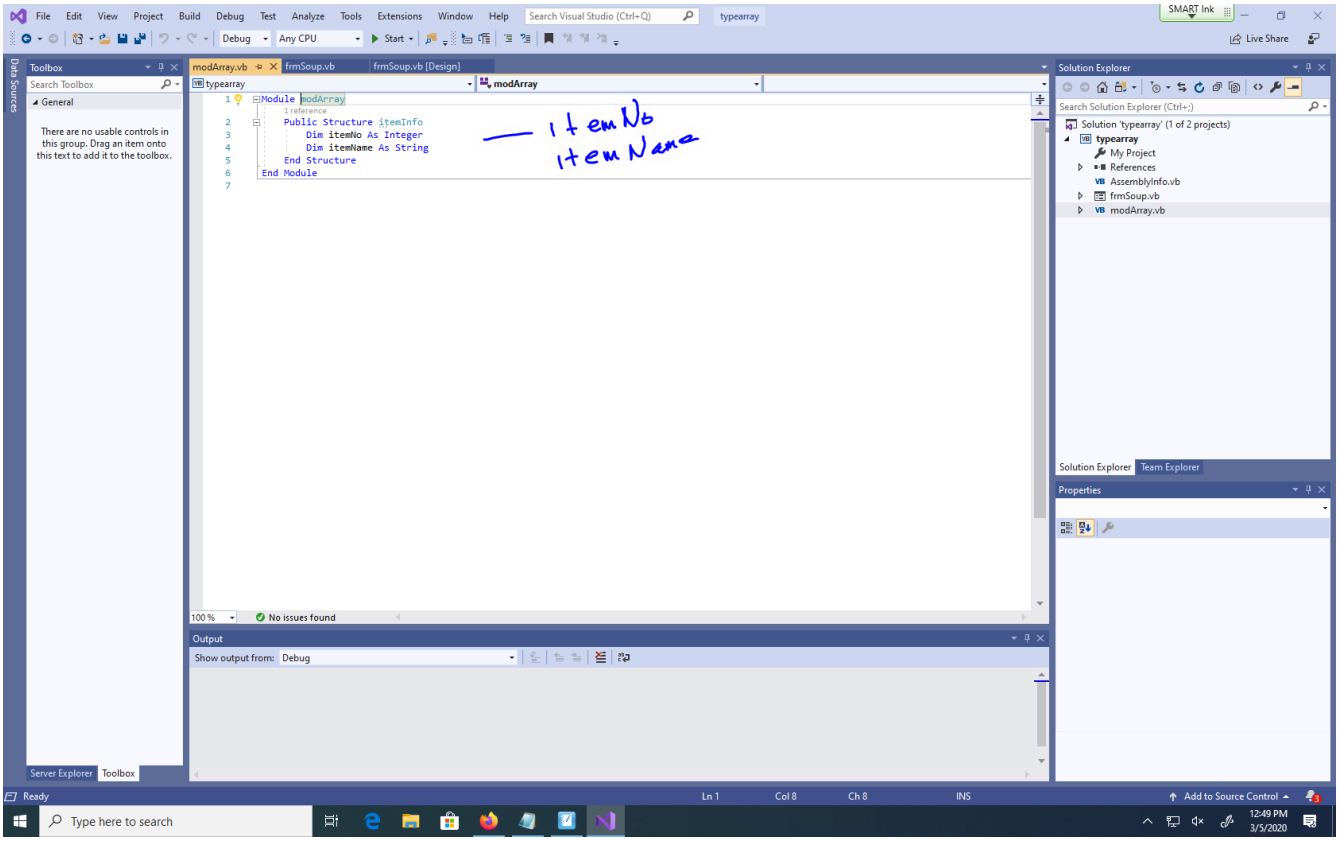

Testing ReDim
Amount:
Start Gather
Gather Amount
Clear
Entries: 4
12
15
17
27
Continue
Second Group of Entries:
Continue Gather
Clear Second
Entries: 6
36
52
Show Contents of Array
12
15
17
27
36
52

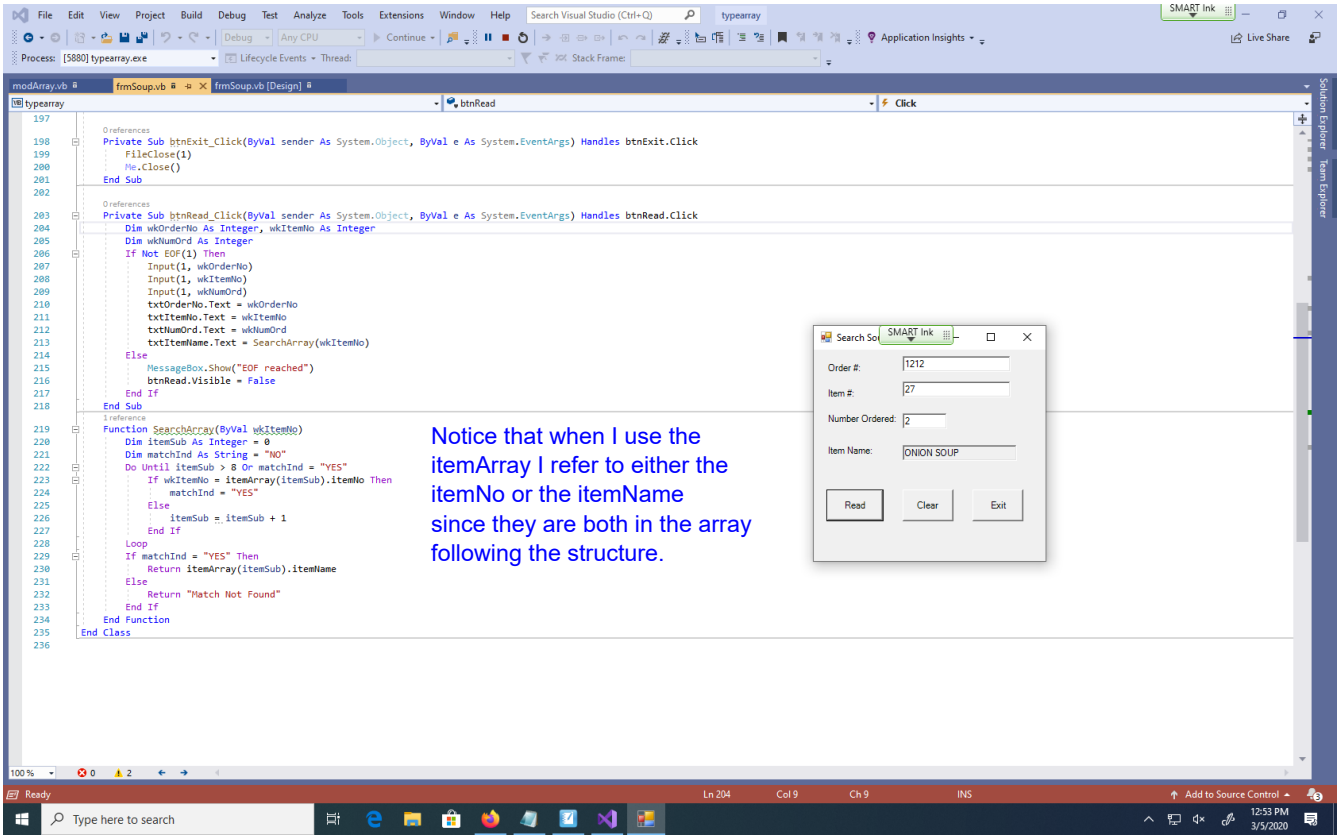

Testing ReDim
Amount:
Start Gather
Gather Amount
Clear
Entries: 4
12
15
17
27
Continue
Second Group of Entries:
Continue Gather
Clear Second
Entries: 6
36
52
Show Contents of Array
12
15
17
27
36
52

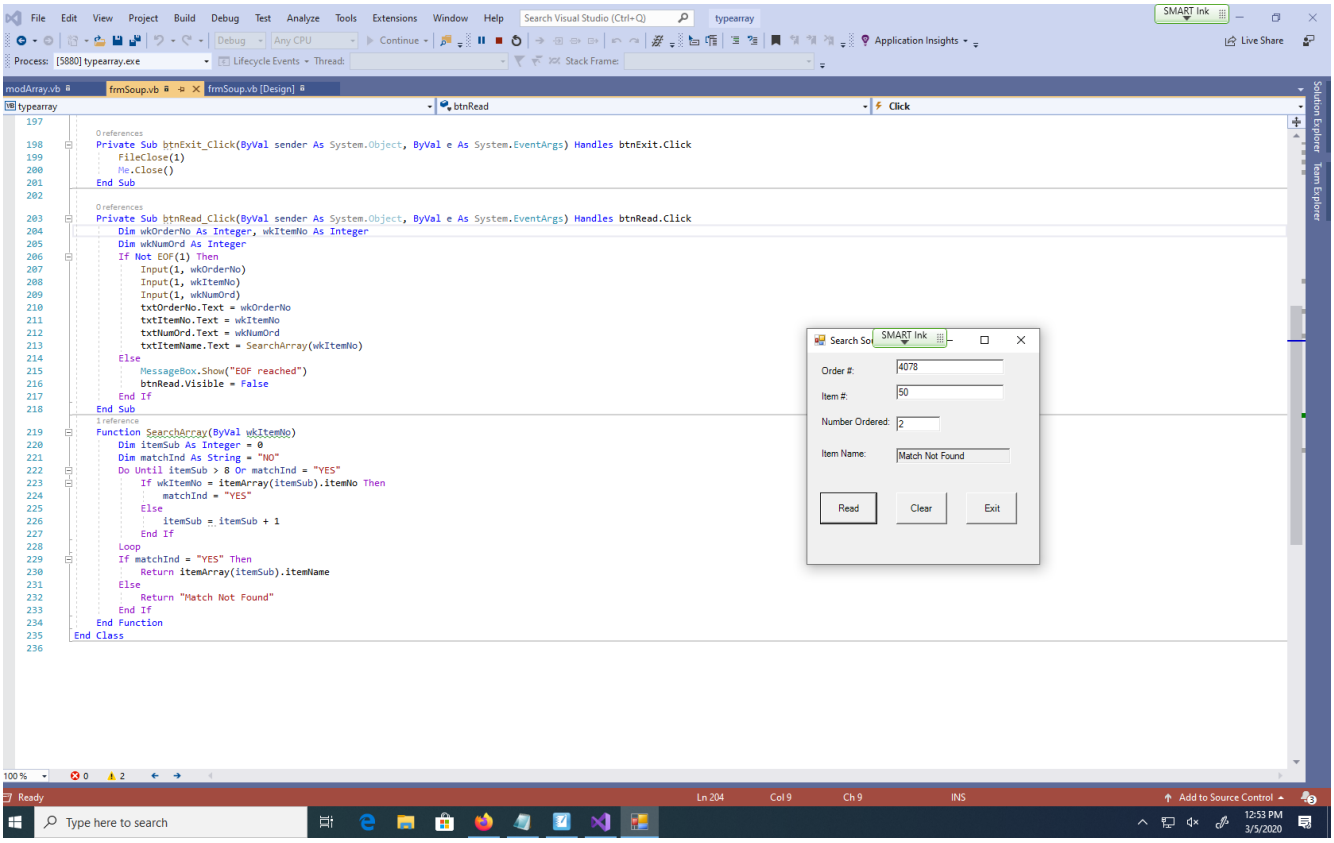

Module modArray
Public Structure itemInfo
Dim itemNo As Integer
Dim itemName As String
End Structure
End Module
I decided to set up the structure in a module. I could have done it in the program.
The structure I set up is itemInfo and anything with that structure will include an itemNo and an itemName.
When I set up the array, I made it itemInfo which means it has an itemNo followed by an itemName.

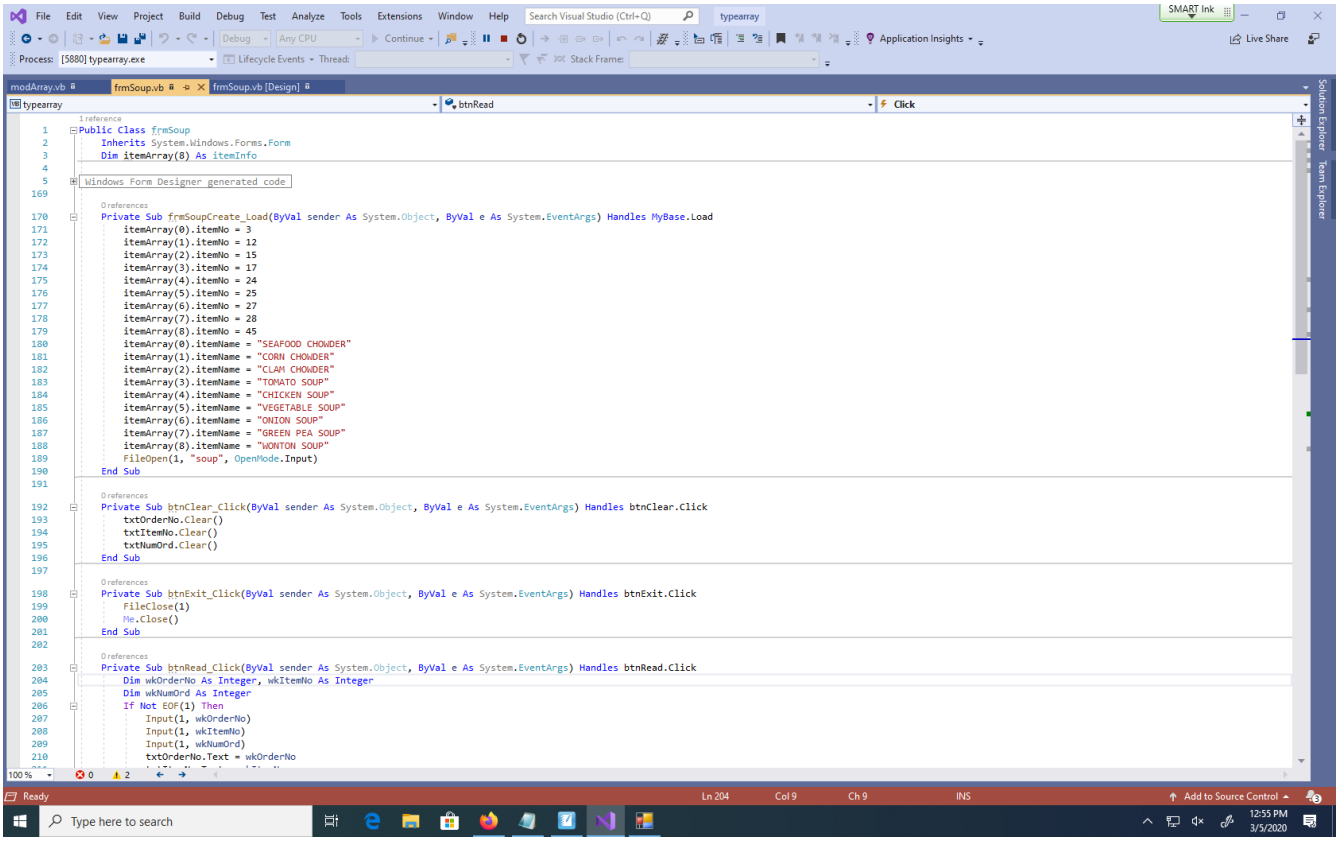



```
Public Class frmSoup
    Inherits System.Windows.Forms.Form
    Dim itemArray(8) As itemInfo

    Windows Form Designer generated code

    Private Sub frmSoupCreate_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        itemArray(0).itemNo = 3
        itemArray(1).itemNo = 12
        itemArray(2).itemNo = 15
        itemArray(3).itemNo = 17
        itemArray(4).itemNo = 24
        itemArray(5).itemNo = 25
        itemArray(6).itemNo = 27
        itemArray(7).itemNo = 28
        itemArray(8).itemNo = 45
        itemArray(0).itemName = "SEAFOOD CHOWDER"
        itemArray(1).itemName = "CORN CHOWDER"
        itemArray(2).itemName = "CLAM CHOWDER"
        itemArray(3).itemName = "TOMATO SOUP"
        itemArray(4).itemName = "CHICKEN SOUP"
        itemArray(5).itemName = "VEGETABLE SOUP"
        itemArray(6).itemName = "ONION SOUP"
        itemArray(7).itemName = "GREEN PEA SOUP"
        itemArray(8).itemName = "WONTON SOUP"
        FileOpen(1, "soup", OpenMode.Input)
    End Sub

    Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClear.Click
        txtOrderNo.Clear()
        txtItemNo.Clear()
        txtNumOrd.Clear()
    End Sub

    Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnExit.Click
        FileClose(1)
        Me.Close()
```

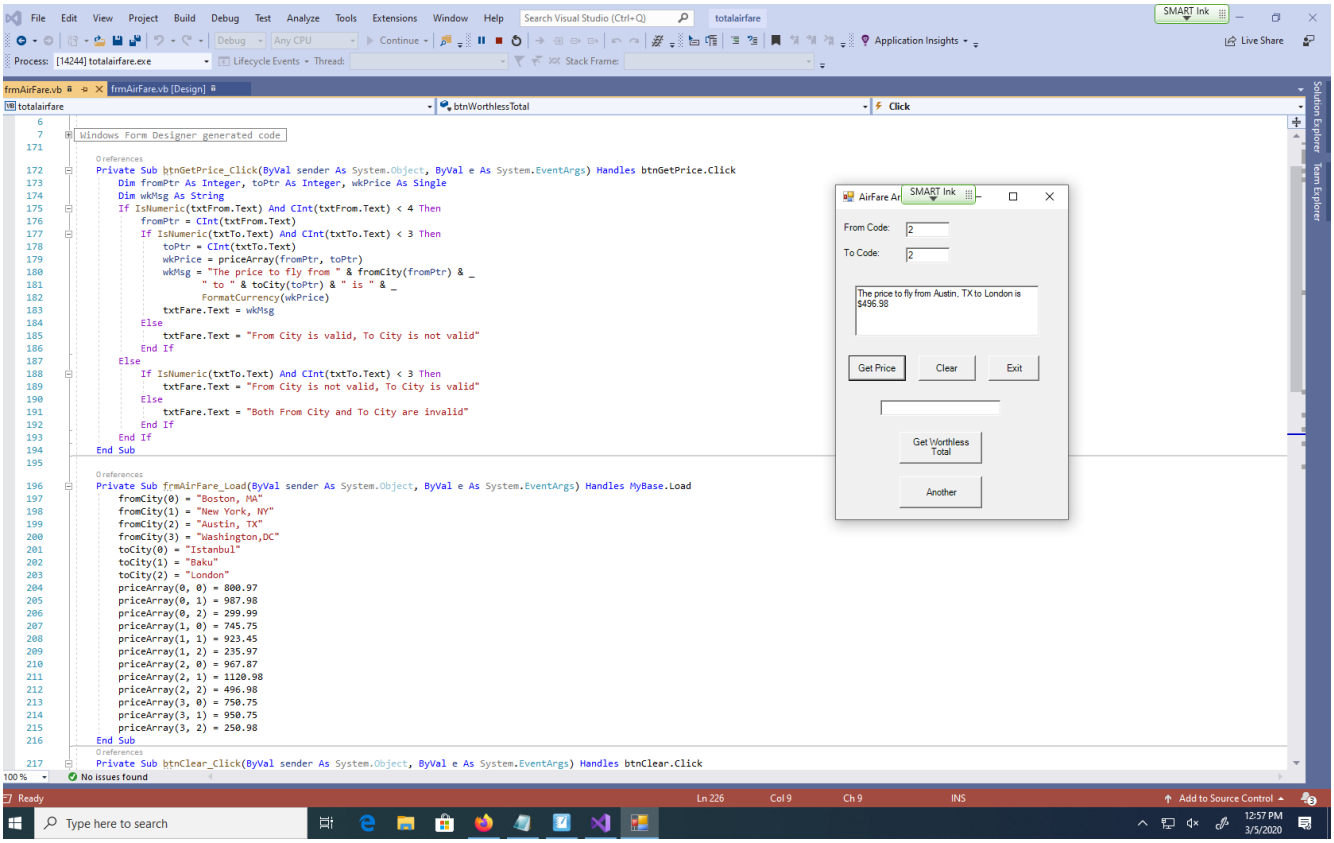

Module modArray
Public Structure itemInfo
Dim itemNo As Integer
Dim itemName As String
End Structure
End Module
itemNo
itemName

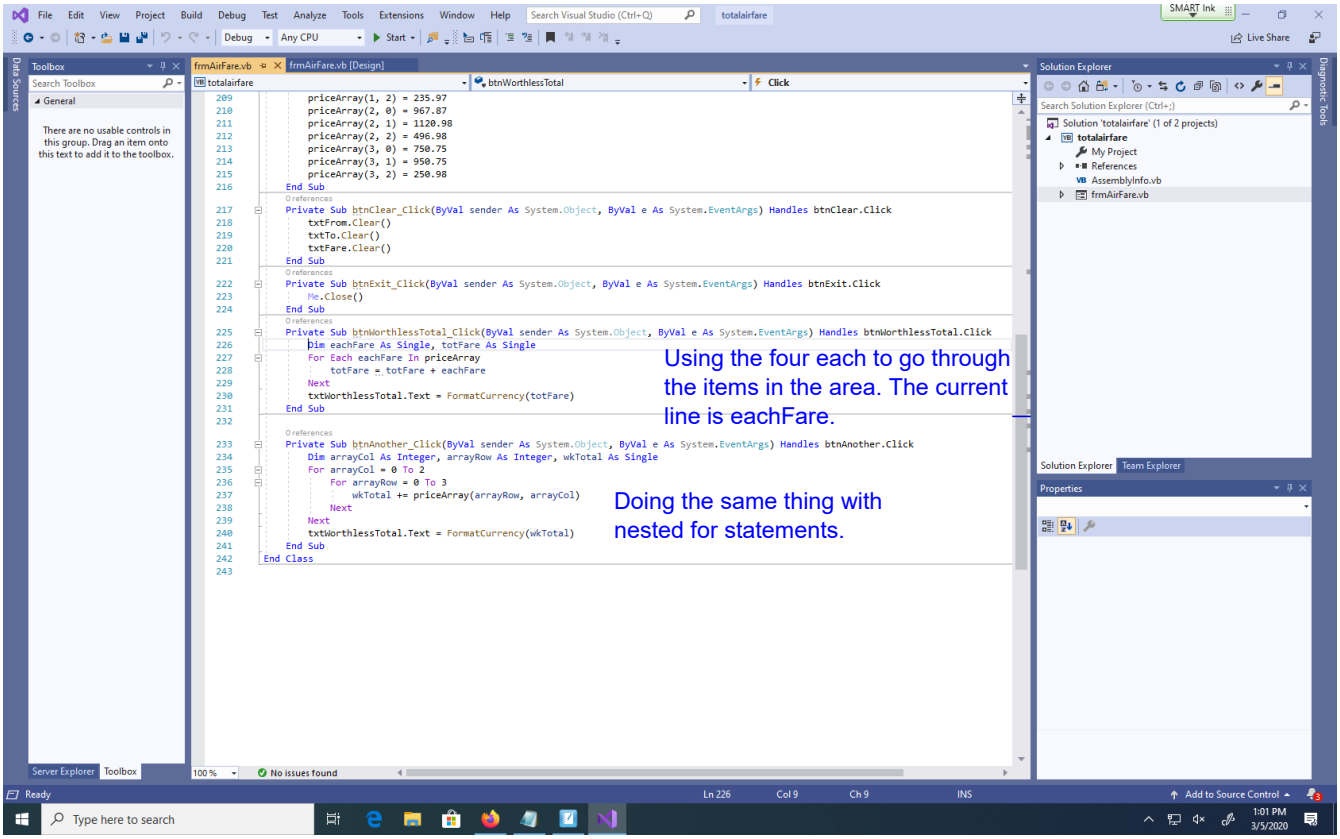

Notice that when I use the itemArray I refer to either the itemNo or the itemName since they are both in the array following the structure.
Order #:
1212
Item #:
27
Number Ordered:
2
Item Name:
ONION SOUP
Read
Clear
Exit

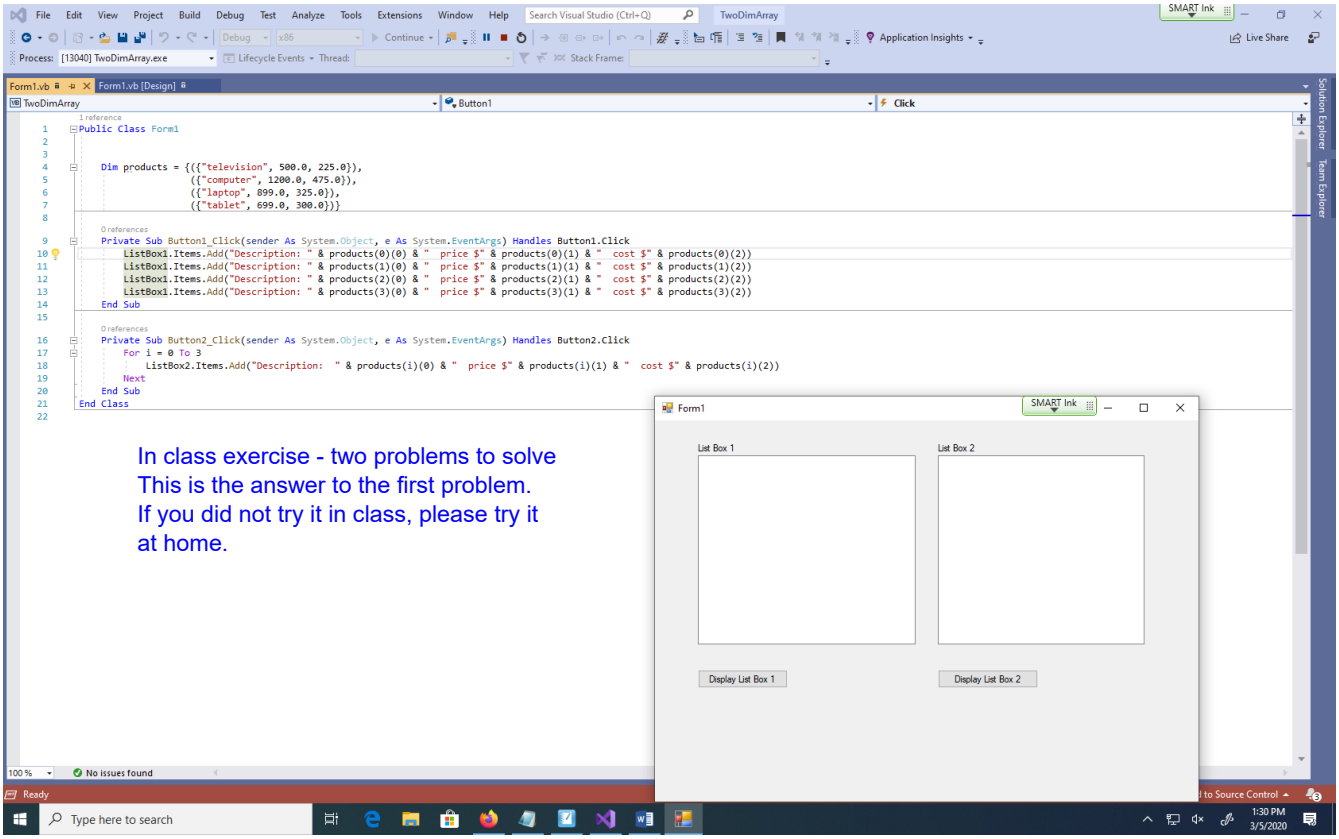

Search Visual Studio (Ctrl+Q)
typearray
modArray.vb
frmSoup.vb
frmSoup.vb [Design]
typearray
btnRead
Click
Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnExit.Click
FileClose(1)
Me.Close()
End Sub
Private Sub btnRead_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRead.Click
Dim wkOrderNo As Integer, wkItemNo As Integer
Dim wkNumOrd As Integer
If Not EOF(1) Then
Input(1, wkOrderNo)
Input(1, wkItemNo)
Input(1, wkNumOrd)
txtOrderNo.Text = wkOrderNo
txtItemNo.Text = wkItemNo
txtNumOrd.Text = wkNumOrd
txtItemName.Text = SearchArray(wkItemNo)
Else
MessageBox.Show("EOF reached")
btnRead.Visible = False
End If
End Sub
Function SearchArray(ByVal wkItemNo)
Dim itemSub As Integer = 0
Dim matchInd As String = "NO"
Do Until itemSub > 8 Or matchInd = "YES"
If wkItemNo = itemArray(itemSub).itemNo Then
matchInd = "YES"
Else
itemSub = itemSub + 1
End If
Loop
If matchInd = "YES" Then
Return itemArray(itemSub).itemName
Else
Return "Match Not Found"
End If
End Function
End Class
Search So
SMART Ink
Order #:
4078
Item #:
50
Number Ordered:
2
Item Name:
Match Not Found
Read
Clear
Exit
Ready
Ln 204
Col 9
Ch 9
INS
Add to Source Control
Type here to search
12:53 PM
3/5/2020



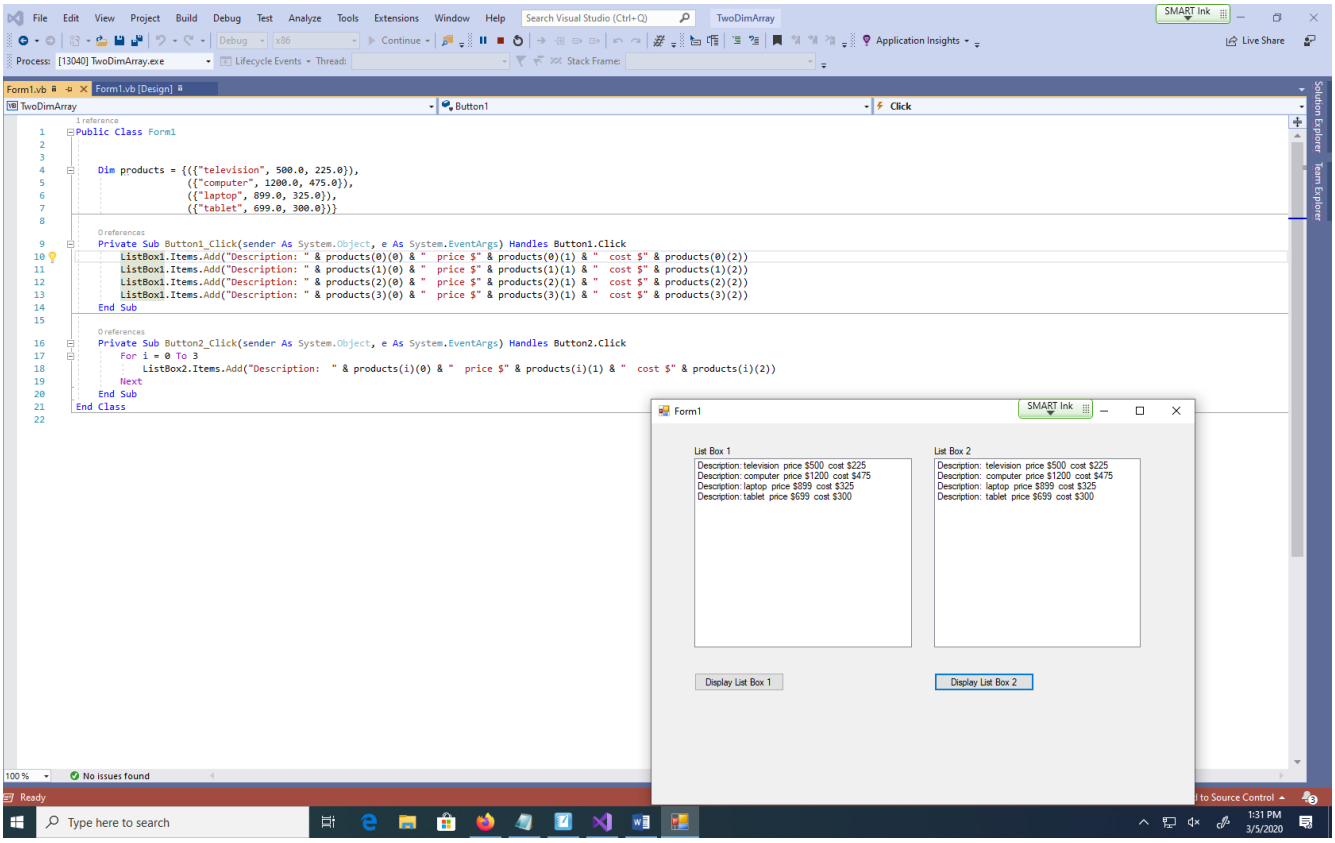
```
1 reference
Public Class frmSoup
    Inherits System.Windows.Forms.Form
    Dim itemArray(8) As itemInfo

    Windows Form Designer generated code

    0 references
    Private Sub frmSoupCreate_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        itemArray(0).itemNo = 3
        itemArray(1).itemNo = 12
        itemArray(2).itemNo = 15
        itemArray(3).itemNo = 17
        itemArray(4).itemNo = 24
        itemArray(5).itemNo = 25
        itemArray(6).itemNo = 27
        itemArray(7).itemNo = 28
        itemArray(8).itemNo = 45
        itemArray(0).itemName = "SEAFOOD CHOWDER"
        itemArray(1).itemName = "CORN CHOWDER"
        itemArray(2).itemName = "CLAM CHOWDER"
        itemArray(3).itemName = "TOMATO SOUP"
        itemArray(4).itemName = "CHICKEN SOUP"
        itemArray(5).itemName = "VEGETABLE SOUP"
        itemArray(6).itemName = "ONION SOUP"
        itemArray(7).itemName = "GREEN PEA SOUP"
        itemArray(8).itemName = "WONTON SOUP"
        FileOpen(1, "soup", OpenMode.Input)
    End Sub

    0 references
    Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClear.Click
        txtOrderNo.Clear()
        txtItemNo.Clear()
        txtNumOrd.Clear()
    End Sub

    0 references
    Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnExit.Click
        FileClose(1)
        Me.Close()
    End Sub

    0 references
    Private Sub btnRead_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRead.Click
        Dim wkOrderNo As Integer, wkItemNo As Integer
        Dim wkNumOrd As Integer
        If Not EOF(1) Then
            Input(1, wkOrderNo)
            Input(1, wkItemNo)
            Input(1, wkNumOrd)
            txtOrderNo.Text = wkOrderNo
```


Windows Form Designer generated code
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)
If IsNumeric(txtTo.Text) And CInt(txtTo.Text) < 3 Then
toPtr = CInt(txtTo.Text)
wkPrice = priceArray(fromPtr, toPtr)
wkMsg = "The price to fly from " & fromCity(fromPtr) & _
" to " & toCity(toPtr) & " is " & _
FormatCurrency(wkPrice)
txtFare.Text = wkMsg
Else
txtFare.Text = "From City is valid, To City is not valid"
End If
Else
If IsNumeric(txtTo.Text) And CInt(txtTo.Text) < 3 Then
txtFare.Text = "From City is not valid, To City is valid"
Else
txtFare.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) = "Baku"
toCity(2) = "London"
priceArray(0, 0) = 800.97
priceArray(0, 1) = 987.98
priceArray(0, 2) = 299.99
priceArray(1, 0) = 745.75
priceArray(1, 1) = 923.45
priceArray(1, 2) = 235.97
priceArray(2, 0) = 967.87
priceArray(2, 1) = 1120.98
priceArray(2, 2) = 496.98
priceArray(3, 0) = 750.75
priceArray(3, 1) = 950.75
priceArray(3, 2) = 250.98
End Sub
Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClear.Click
AirFare Ar
SMART Ink
From Code:
2
To Code:
2
The price to fly from Austin, TX to London is $496.98
Get Price
Clear
Exit
Get Worthless Total
Another


Using the four each to go through the items in the area. The current line is eachFare.
Doing the same thing with nested for statements.



In class exercise - two problems to solve
This is the answer to the first problem.
If you did not try it in class, please try it at home.


Public Class Form1
Dim products = {{{"television", 500.0, 225.0}},
{{"computer", 1200.0, 475.0}},
{{"laptop", 899.0, 325.0}},
{{"tablet", 699.0, 300.0}}}
Private Sub Button1_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click
ListBox1.Items.Add("Description: " & products(0)(0) & " price $" & products(0)(1) & " cost $" & products(0)(2))
ListBox1.Items.Add("Description: " & products(1)(0) & " price $" & products(1)(1) & " cost $" & products(1)(2))
ListBox1.Items.Add("Description: " & products(2)(0) & " price $" & products(2)(1) & " cost $" & products(2)(2))
ListBox1.Items.Add("Description: " & products(3)(0) & " price $" & products(3)(1) & " cost $" & products(3)(2))
End Sub
Private Sub Button2_Click(sender As System.Object, e As System.EventArgs) Handles Button2.Click
For i = 0 To 3
ListBox2.Items.Add("Description: " & products(i)(0) & " price $" & products(i)(1) & " cost $" & products(i)(2))
Next
End Sub
End Class
Form1
List Box 1
Description: television price $500 cost $225
Description: computer price $1200 cost $475
Description: laptop price $899 cost $325
Description: tablet price $699 cost $300
List Box 2
Description: television price $500 cost $225
Description: computer price $1200 cost $475
Description: laptop price $899 cost $325
Description: tablet price $699 cost $300
Display List Box 1
Display List Box 2