```
Public Function IsArrayEmpty(Arr As Variant) As Boolean
'helper function - checks if an array is empty. Returns True if empty
    Dim N As Long
    On Error Resume Next
    Err.Clear
    N = LBound(Arr)
    If Err.Number = 0 Then
        IsArrayEmpty = False
    Else
        IsArrayEmpty = True
    End If
End Function


Function FindLastRow(ByVal sheetName As String) As Integer
    'find the last row with data entered by checking the column with pier diameter

    Dim wks As Worksheet
    Dim Last As Integer

    Set wks = ActiveWorkbook.Worksheets(sheetName)

    With wks
        Last = .Cells(.Rows.Count, "I").End(xlUp).Row
    End With

    FindLastRow = Last

End Function

Sub SortProjects(ByVal sheetName As String)
    'Sorts projects first by 3-letter code, then by number

    Dim wks As Worksheet
    Set wks = ActiveWorkbook.Worksheets(sheetName)

    'Find the last filled-in row
    lastRow = FindLastRow(sheetName)

    'Activates the database worksheet
    wks.Activate

    'Selects and sorts the data
    ActiveSheet.Range("A7", "AW" & CStr(lastRow)).Select
    Selection.Sort _
    key1:=ActiveSheet.Range("D6"), order1:=xlAscending, _
    Key2:=ActiveSheet.Range("E6"), Order2:=xlAscending, _
    Header:=xlNo, OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom, _
        DataOption1:=xlSortNormal

    'selects a random cell to unselect the rest
    ActiveSheet.Range("A20").Select
    Application.CutCopyMode = False

    'Number the entries
    Dim Counter As Integer
    Counter = 0
    For Each MyCell In ActiveSheet.Range("A7:A" & lastRow)
        Counter = Counter + 1
        MyCell.Value = Counter
    Next MyCell


End Sub
Function getSoilInfo(FileName, Depth, FFE)
    'function to open soil profile workbook, separate upper/lower zone, extract and return N-values
    Dim currWB As Workbook

    Dim wbTarget As Workbook
    Dim CloseIt As Boolean
    Dim nVal As Boolean
    Dim limit As Integer
    Dim name As String
    name = FileName

    Set currWB = ActiveWorkbook
    Set TOHinput = currWB.Worksheets("Input").Range("E10")
    If IsEmpty(TOHinput) Then
        MsgBox "You must manually input TOH elevations"
    End If
    TOHs = Split(TOHinput.Value, ",") 'turns TOH values into array

    For i = LBound(TOHs) To UBound(TOHs)
        TOHs(i) = CInt(TOHs(i))
    Next i
    On Error Resume Next
    Set wbTarget = Workbooks(FileName)




    '~~> Check and make sure workbook was opened
    If Err.Number = 1004 Then
        MsgBox "Soil Profile file does not exist. Check for typos."
        Exit Function
    End If

    On Error GoTo 0

    Set graph = wbTarget.Worksheets("Graphs")

    nVal = True
    'case: no N-values recorded
    If IsEmpty(graph.Range("A4")) Then
        'Case: qu values recorded
        nVal = False
        If Not IsEmpty(graph.Range("A168")) Then

            multiArr = obtainStressVals(name, nVal, FFE, TOHs, Depth)
        'Case: no n-values or qu values
        Else
            MsgBox "Sorry, no N-values or qu were found. Data must be entered manually."
            getSoilInfo = 0
        End If
    'case: N-values recorded
    Else
        multiArr = obtainStressVals(name, nVal, FFE, TOHs, Depth)
    End If

    If CloseIt = True Then
        '~~> If the target workbook was opened by the macro, close it
        wbTarget.Close savechanges:=False
    Else
        '~~> If the target workbook was already open, reactivate this workbook
        ThisWorkbook.Activate
    End If

    getSoilInfo = multiArr

End Function

Function obtainStressVals(FileName, nVal As Boolean, FFE, TOHs, Depth)
    'helper function for getSoilInfo
    On Error Resume Next
    Set wbTarget = Workbooks(FileName)


    '~~> Check and make sure workbook was opened
    If Err.Number = 1004 Then
        MsgBox "File does not exist!"
        Exit Function
    End If

    On Error GoTo 0
    Set graph = wbTarget.Worksheets("Graphs")

    Dim StartNumber As Integer
    If nVal Then
        StartNumber = 4
        EndNumber = 30
    Else
        StartNumber = 168
        EndNumber = 198
    End If

    Dim cel As Range

    Dim upLim As Variant
    If Not FFE = 0 Then 'if there is an FFE value
        For i = LBound(TOHs) To UBound(TOHs)
            If TOHs(i) > FFE Then
                TOHs(i) = FFE
            End If
        Next i
    End If


    'arrays for the two zones

    Dim upperZone() As Variant, upSize As Integer
    upSize = -1
    Dim lowerZone() As Variant, lowSize As Integer
    lowSize = -1
    Dim countCol As Integer
    countCol = 0


    'obtains N-values or qu for upper and lower zones
    'Must use ReDim and Preserve because arrays are dynamically sized
    For i = graph.Range("A" & CStr(StartNumber)).Column To graph.Range("BH" & CStr(EndNumber)).Column
Step 2
        If Not IsEmpty(graph.Cells(4, i)) Then
            upLim = TOHs(countCol) 'obtain surface elevation or FFE
            limit = upLim - Depth 'separate upper and lower zones
        Else
            upLim = 0
            limit = 0
        End If
        For Counter = StartNumber To EndNumber

            Set cel = graph.Cells(Counter, i)

            If IsEmpty(cel) Then
                Exit For
            End If



            If Not IsEmpty(cel.Offset(columnOffset:=1)) Then
                Dim useVal As Double 'the n-value or qu value
                useVal = cel.Offset(columnOffset:=1).Value
                If Not nVal Then 'if qu
                    useVal = useVal * 2  'converting from tsf to ksf
                Else 'if n-value
                    If Not useVal = Int(useVal) Then   'Case for value such as 50/3
                        useVal = 50
                    End If
                End If

                'determine if value belongs to upper or lower zone
                If cel.Value > limit And cel.Value < upLim Then
                    upSize = upSize + 1
                    ReDim Preserve upperZone(upSize)
                    upperZone(UBound(upperZone)) = CStr(useVal)

                ElseIf cel.Value < limit And cel.Value < upLim Then
                    lowSize = lowSize + 1
                    ReDim Preserve lowerZone(lowSize)
                    lowerZone(UBound(lowerZone)) = CStr(useVal)

                End If

            End If
        Next Counter
        countCol = countCol + 1
    Next i
    'combines zones into single multidimensional array to return
    Dim multiArr(2) As Variant
    multiArr(0) = upperZone
    multiArr(1) = lowerZone
    multiArr(2) = nVal
    If CloseIt = True Then
        '~~> If the target workbook was opened by the macro, close it
        wbTarget.Close savechanges:=False
    Else
        '~~> If the target workbook was already open, reactivate this workbook
        ThisWorkbook.Activate
    End If
    obtainStressVals = multiArr
End Function



Public Function SrtViaWorksheet(List, WorkBookToUse)
'Not written by me, function to sort an array in ascending order
    Dim WS As Worksheet ' temporary worksheet
    Dim R As Range
    Dim N As Long

    Application.ScreenUpdating = False

    ' create a new sheet
    Set WS = WorkBookToUse.Worksheets.Add

    ' put the array values on the worksheet
    Set R = WS.Range("A1").Resize(UBound(List) - LBound(List) + 1, 1)
    R = Application.Transpose(List)

    ' sort the range
    R.Sort key1:=R, order1:=xlAscending, MatchCase:=False

    ' load the worksheet values back into the array
    For N = 0 To R.Rows.Count - 1
        List(N) = R(N + 1, 1)
    Next N

    ' delete the temporary sheet
    Application.DisplayAlerts = False
    WS.Delete
    Application.DisplayAlerts = True
    Application.ScreenUpdating = True


    SrtViaWorksheet = List
End Function


Sub logModDataSub()
    'main sub-takes data from Input sheet and puts it into Database Sheet
    'calls helper functions

    Dim inputSheet As Worksheet 'input
    Dim databaseSheet As Worksheet 'database
    Dim MTD As Workbook 'this workbook
    Dim nextRow As Integer
    Dim projYear As String
    Dim projCode As String
    Dim projNum As String
    Dim fullCode As String
    Dim HS As Double
    Dim DF As Double
    Set MTD = ActiveWorkbook
    Set inputSheet = MTD.Worksheets("Input")
    Set databaseSheet = MTD.Worksheets("MASTER Modulus Test Database")
    HS = inputSheet.Range("B14").Value
    DF = inputSheet.Range("B15").Value
    nextRow = FindLastRow("MASTER Modulus Test Database") + 1 'finds the next unfilled row


    'get soil profile info
    Dim soilName As String
    Dim FFE As Variant
    If Not IsEmpty(inputSheet.Range("E11")) Then
        FFE = inputSheet.Range("E11").Value
    Else
        FFE = 0
    End If


    Dim pierDepth As Variant
    pierDepth = HS + DF
    checkSoil = inputSheet.Range("E9").Value
    If Not IsEmpty(checkSoil) Then
        soilArray = getSoilInfo(checkSoil, pierDepth, FFE)
         Dim upHigh As Variant
         Dim upLow As Variant
         Dim upRep As Variant
         Dim lowHigh As Variant
         Dim lowLow As Variant
         Dim nVal As Boolean

         upperZone = soilArray(0)
         lowerZone = soilArray(1)
         nVal = soilArray(2)

         If Not IsArrayEmpty(upperZone) Then
             upperZone = SrtViaWorksheet(upperZone, MTD) 'sort in ascending order
             'assign high,low,rep
             upHigh = upperZone(UBound(upperZone))
             upLow = upperZone(LBound(upperZone))
             If UBound(upperZone) >= 2 Then
                upRep = upperZone(((UBound(upperZone) + 1) \ 3) - 1)
                If UBound(upperZone) Mod 2 = 0 Then
                    upRep = upRep + 1
                End If
             Else
                upRep = upLow
             End If
             'enter in worksheet
             If nVal Then
                databaseSheet.Range("M" & CStr(nextRow)).Value = upHigh
                databaseSheet.Range("N" & CStr(nextRow)).Value = upLow
                databaseSheet.Range("O" & CStr(nextRow)).Value = upRep
             Else
                databaseSheet.Range("Q" & CStr(nextRow)).Value = upHigh
                databaseSheet.Range("R" & CStr(nextRow)).Value = upLow
                databaseSheet.Range("P" & CStr(nextRow)).Value = upRep
             End If
         Else
             MsgBox "There are no stress values in the upper zone"
         End If

        If Not IsArrayEmpty(lowerZone) Then
             lowerZone = SrtViaWorksheet(lowerZone, MTD) 'sort ascending order
             'assign high, low, rep
             lowHigh = lowerZone(UBound(lowerZone))
             lowLow = lowerZone(LBound(lowerZone))
             If UBound(lowerZone) >= 2 Then
                lowRep = lowerZone(((UBound(lowerZone) + 1) \ 3) - 1)
                If UBound(lowerZone) Mod 2 = 0 Then
                    lowRep = lowRep + 1
                End If
```

```
            Else
               lowRep = lowLow
            End If

            'enter in worksheet
            If nVal Then
               databaseSheet.Range("U" & CStr(nextRow)).Value = lowRep
               databaseSheet.Range("V" & CStr(nextRow)).Value = lowHigh
               databaseSheet.Range("W" & CStr(nextRow)).Value = lowLow
            Else
               databaseSheet.Range("X" & CStr(nextRow)).Value = lowRep
               databaseSheet.Range("Y" & CStr(nextRow)).Value = lowHigh
               databaseSheet.Range("Z" & CStr(nextRow)).Value = lowLow
            End If
        Else
            MsgBox "There are no stress values in the lower zone"
        End If
    Else
        MsgBox "You did not upload a soil profile."
    End If

    'modulus test data
    If Not IsEmpty(inputSheet.Range("E13")) Then
        Dim FileName As String
        FileName = inputSheet.Range("E13").Value
        Dim ModTest As Workbook

        Set ModTest = Workbooks(FileName)


       If Err.Number = 1004 Then
            MsgBox "Modulus file does not exist. Check for typos."
       Else
            Dim loadSheet As Worksheet
            Set loadSheet = ModTest.Worksheets("Load Test")
            Dim modSheet As Worksheet
            Set modSheet = ModTest.Worksheets("Modulus")

            Dim DesignStress As Integer
            Dim KgDesign As Integer
            Dim Kg100 As Integer
            Dim Kg150 As Integer
            Dim Deflect100 As Double
            Dim Deflect150 As Double
            Dim tell100 As Double
            Dim tell150 As Double


            If Not IsEmpty(loadSheet.Range("I75")) Then
                DesignStress = loadSheet.Range("I75").Value
                databaseSheet.Range("AE" & CStr(nextRow)).Value = DesignStress
            End If
            If Not IsEmpty(modSheet.Range("I6")) Then
                KgDesign = modSheet.Range("I6").Value
                databaseSheet.Range("AB" & CStr(nextRow)).Value = KgDesign
            End If
            If Not IsEmpty(modSheet.Range("F22")) Then
                Kg100 = modSheet.Range("F22").Value
                databaseSheet.Range("AC" & CStr(nextRow)).Value = Kg100
            End If
            If Not IsEmpty(modSheet.Range("F25")) Then
                Kg150 = modSheet.Range("F25").Value
                databaseSheet.Range("AD" & CStr(nextRow)).Value = Kg150
            End If
            If Not IsEmpty(modSheet.Range("D22")) Then
                Deflect100 = Abs(modSheet.Range("D22").Value)
                databaseSheet.Range("AF" & CStr(nextRow)).Value = Deflect100
            End If
            If Not IsEmpty(modSheet.Range("D25")) Then
                Deflect150 = Abs(modSheet.Range("D25").Value)
                databaseSheet.Range("AH" & CStr(nextRow)).Value = Deflect150
            End If
            If Not IsEmpty(modSheet.Range("E22")) Then
```

```
                tell100 = Abs(modSheet.Range("E22").Value)
                databaseSheet.Range("AM" & CStr(nextRow)).Value = tell100
            End If
            If Not IsEmpty(modSheet.Range("E25")) Then
                tell150 = Abs(modSheet.Range("E25").Value)
                databaseSheet.Range("AN" & CStr(nextRow)).Value = tell150
            End If
            If Not IsEmpty(loadSheet.Range("G4")) Then
                fullCode = loadSheet.Range("G4").Value
            Else
                fullCode = inputSheet.Range("B8").Value
            End If
            If Not IsEmpty(loadSheet.Range("G2")) Then
                Dim Index As Integer
                Location = loadSheet.Range("G2").Value
                Index = InStr(Location, ",")
                If Not Index = 0 Then

                    City = Left(Location, Index - 1)
                    State = Right(Location, Len(Location) - Index)
                    databaseSheet.Range("G" & CStr(nextRow)).Value = City
                    databaseSheet.Range("H" & CStr(nextRow)).Value = State
                End If
            End If
            If Not IsEmpty(loadSheet.Range("G1")) Then
                databaseSheet.Range("F" & CStr(nextRow)).Value = loadSheet.Range("G1").Value 'project
name
            End If
            If Not IsEmpty(loadSheet.Range("I76")) Then
                databaseSheet.Range("I" & CStr(nextRow)).Value = loadSheet.Range("I76").Value 'project
 name
            End If

            'full code
            projYear = Left(fullCode, 3)
            projCode = Mid(fullCode, 5, 3)
            projNum = Right(fullCode, 5)

            databaseSheet.Range("C" & CStr(nextRow)).Value = projYear 'P17
            databaseSheet.Range("D" & CStr(nextRow)).Value = projCode 'PMW
            databaseSheet.Range("E" & CStr(nextRow)).Value = projNum '00900
            databaseSheet.Range("AQ" & CStr(nextRow)).Value = fullCode 'full code

            'check if telltale values are positive
            Dim Check As Boolean
            Check = False
            For Each MyCell In modSheet.Range("E16", "E29")
                If MyCell.Value > 0 Then
                    Check = True
                End If
            Next MyCell
            If Check = True Then
                MsgBox "Warning:Some of the tell-tale values are positive."
            End If

        End If

    Else
        fullCode = inputSheet.Range("B8").Value
        projYear = Left(fullCode, 3)
        projCode = Mid(fullCode, 5, 3)
        projNum = Right(fullCode, 5)

        databaseSheet.Range("C" & CStr(nextRow)).Value = projYear 'P17
        databaseSheet.Range("D" & CStr(nextRow)).Value = projCode 'PMW
        databaseSheet.Range("E" & CStr(nextRow)).Value = projNum '00900
        databaseSheet.Range("AQ" & CStr(nextRow)).Value = fullCode 'full code
    End If



    'input values to next row of database sheet
    If Not IsEmpty(inputSheet.Range("B9")) Then
        databaseSheet.Range("F" & CStr(nextRow)).Value = inputSheet.Range("B9").Value 'name
    End If
    If Not IsEmpty(inputSheet.Range("B10")) Then
    databaseSheet.Range("G" & CStr(nextRow)).Value = inputSheet.Range("B10").Value 'city
    End If
    If Not IsEmpty(inputSheet.Range("B11")) Then
    databaseSheet.Range("H" & CStr(nextRow)).Value = inputSheet.Range("B11").Value 'state
    End If
    If Not IsEmpty(inputSheet.Range("B12")) Then
    databaseSheet.Range("I" & CStr(nextRow)).Value = inputSheet.Range("B12").Value 'diameter
    End If
    databaseSheet.Range("J" & CStr(nextRow)).Value = inputSheet.Range("B13").Value 'system
    databaseSheet.Range("K" & CStr(nextRow)).Value = HS 'hs
    databaseSheet.Range("L" & CStr(nextRow)).Value = DF 'df
    databaseSheet.Range("AR" & CStr(nextRow)).Value = inputSheet.Range("B16").Value 'aggregate
    databaseSheet.Range("AP" & CStr(nextRow)).Value = inputSheet.Range("B17").Value 'hammer
    databaseSheet.Range("AS" & CStr(nextRow)).Value = inputSheet.Range("B18").Value 'BST
    databaseSheet.Range("B" & CStr(nextRow)).Value = Date 'date

    Dim qgInflect As Double
    Dim defInflect As Double
    qgInflect = Abs(inputSheet.Range("B20").Value)
    defInflect = Abs(inputSheet.Range("B21").Value)

    databaseSheet.Range("AG" & CStr(nextRow)).Value = inputSheet.Range("B19").Value 'IP reached
    databaseSheet.Range("AI" & CStr(nextRow)).Value = qgInflect 'qg inflect
    databaseSheet.Range("AJ" & CStr(nextRow)).Value = defInflect 'def inflect
    databaseSheet.Range("AL" & CStr(nextRow)).Value = inputSheet.Range("B22").Value 'telltale at IP
    databaseSheet.Range("AK" & CStr(nextRow)).Value = (qgInflect * 1000) / 144 / defInflect 'KG at IP
calculation
    databaseSheet.Range("AO" & CStr(nextRow)).Value = inputSheet.Range("B23").Value 'behavior

    'delete soil input
    inputSheet.Range("E9:E13").ClearContents
    inputSheet.Range("B8:B23").ClearContents


    'sort the worksheet - Do this last because it changes active worksheet
    Call SortProjects("MASTER Modulus Test Database")
    MsgBox "Check the soil profile info and enter rest of data manually. The new data is highlighted."

    'highlight the new line of data

    Set newCell = databaseSheet.Range("AQ1", "AQ" & nextRow).Find(fullCode)
    databaseSheet.Rows(newCell.Row).Select

End Sub
```