Appendix F

Service Life Model -Visual Basic Program Code

About Form

```
Option Explicit


Private Sub CommandButton1_Click()
    Unload Me
End Sub


'Private Sub UserForm_Initialize()
    'Dim strcopyright As String
    'strcopyright = Chr(169)
    'Label2.Caption = "Copyright " & strcopyright & " 2003"
    'Label2.Caption = "Copyright what's wrong"
'End Sub
Private Sub UserForm_Click()


End Sub
```

D_c Calculation

```
Option Explicit

Private Sub cmdcancel_Click()
    Unload Me
End Sub

Private Sub cmdcomp_Click()

With dccalc
    '.Show
    Application.ScreenUpdating = True
    'if .CancelPressed Then Exit Sub

    Dim unit4x(2) As String
    Dim unit4cxt(2) As String
    Dim unit4dc(2) As String
    unit4x(1) = unit4xSI
    unit4x(2) = unit4xUS
    unit4cxt(1) = unit4cxtSI
    unit4cxt(2) = unit4cxtUS
```

```
    unit4dc(1) = unit4dcSI
    unit4dc(2) = unit4dcUS

'*********************************************************************
**
    '*   Read Co, Dc, x data and bridge label from the 'inputdata' useform   *

'*********************************************************************
**
    '
    Dim bridgename As String, servicelife As Double, specid As String
    Dim rnggcxt As Range, rnggx As Range
    Dim initDc As Double
    If .txtboxbridge.Text = "" Then
        MsgBox "Please enter label of the bridge!"
        Exit Sub
    End If
    bridgename = .txtboxbridge.Text
    If .txtboxid.Text = "" Then
        MsgBox "Please enter ID of the specimen!"
        Exit Sub
    End If
    specid = .txtboxid.Text
    bridgename = bridgename & "-" & specid


    If .TextBoxInitDc.Text = "" Then
        MsgBox "The default initial Dc value is 10 " & unit4dc(unitset) & "!"
    End If
    initDc = .TextBoxInitDc.Text
    If initDc <= 0.001 Then
        MsgBox "Negative and values close to 0 are not good initial values for Dc! The
default value 10 is used here!"
        initDc = 10
    End If


    'If .txtboxt.Text = "" Then
    '    MsgBox "Please enter service years!"
    '    Exit Sub
    'End If
    'servicelife = .txtboxt.Text
    'If servicelife <= 0 Then
    '    MsgBox "Negative or 0 is not valid value for service life!"
    '    Exit Sub
    'End If
```

```
    Dim rngt As Range
    If .RefEditt.Value = "" Then
       MsgBox "Please input concrete age at sampling!"
       Exit Sub
    End If
    Set rngt = Range(.RefEditt.Value)
    If rngt.Columns.Count <> 1 Or rngt.Rows.Count <> 1 Then
       MsgBox "Input concrete age from a single cell in the spreadsheet!"
       Exit Sub
    End If
    servicelife = rngt.Cells(1, 1).Value
    If servicelife <= 0 Then
       MsgBox "Negative or 0 is not valid value for concrete age!"
       Exit Sub
    End If


    Dim c0 As Double
    Dim x0 As Double
    Dim rngc0 As Range, rngx0 As Range
    If .RefEditC0.Value = "" Or .RefEditx0.Value = "" Then
       MsgBox "Please choose the input cell for Co and x0!"
       Exit Sub
    End If
    Set rngc0 = Range(.RefEditC0.Value)
    Set rngx0 = Range(.RefEditx0.Value)
    If rngc0.Columns.Count <> 1 Or rngx0.Columns.Count <> 1 Or rngc0.Rows.Count <>
1 Or rngx0.Rows.Count <> 1 Then
       MsgBox "Input Co/x0 value from single cell in the spreadsheet!"
       Exit Sub
    End If
    c0 = rngc0.Cells(1, 1).Value
    x0 = rngx0.Cells(1, 1).Value
    If c0 <= 0 Then
       MsgBox "Negative or 0 is not valid value for Co!"
       Exit Sub
    End If
    If x0 < 0 Then
       MsgBox "Negative is not valid value for depth x0!"
       Exit Sub
    End If
    'If .txtboxCoatx0.Text = "" Then
    '    MsgBox "Please input value of Co!"
    '    Exit Sub
    'End If
    'If .txtboxx0.Text = "" Then
```

```
'    MsgBox "Please input value of depth x0!"
'    Exit Sub
'End If
'c0 = .txtboxCoatx0.Text
'x0 = .txtboxx0.Text


If .Refit1.Value = "" Or .Refit2.Value = "" Then
   MsgBox "Please choose the input data range!"
   Exit Sub
End If
Set rnggcxt = Range(.Refit1.Value)
Set rnggx = Range(.Refit2.Value)
'Set rnggco = Range(.Refit3.Value)
'First check the data are from 3 columns
If rnggcxt.Columns.Count <> 1 Or rnggx.Columns.Count <> 1 Then
   MsgBox "Check if the data are input from three columns!"
   Exit Sub
End If
If rnggcxt.Count < 3 Or rnggx.Count < 3 Or rnggcxt.Count <> rnggx.Count Then
   MsgBox "Please input a minimum of three pairs of data for Cxt and x!"
   Exit Sub
End If

Dim datanum As Long
Dim calcxt() As Double, calx() As Double
datanum = rnggcxt.Count
   ReDim Preserve calcxt(1 To datanum) As Double
   ReDim Preserve calx(1 To datanum) As Double

'
'Check if select the output range
'If .Refit4.Value = "" Then
'    MsgBox "Please choose the output range!"
'    Exit Sub
'End If
'Set rnggout = Range(.Refit4.Value)
'
'
'
'Save the data from worksheet to the arrays Codata, dcdata and cxtdata
Dim r As Long
For r = 1 To datanum
   calcxt(r) = rnggcxt.Cells(r, 1).Value
   If calcxt(r) = 0 Then
         MsgBox "There are some empty cells within Cxt data, please check!"
```

```
            Exit Sub
        End If
    Next r
    For r = 1 To datanum
        calx(r) = rnggx.Cells(r, 1).Value
        If calx(r) = 0 Then
            MsgBox "There are some empty cells within x data, please check!"
            Exit Sub
        End If
    Next r


    .Hide
    Application.ScreenUpdating = True
    Application.ScreenUpdating = False

    '*************************************************************
    '*              Start calculation!                           *
    '*************************************************************
    '
    AddIns("Solver Add-in").Installed = True
    AddIns("analysis toolpak").Installed = True
    Dim usesheet As Worksheet

    Set usesheet = Worksheets.Add
    usesheet.Range("F1").Value = "Difussion Constant Least Square Estimate(" &
unit4dc(unitset) & ")"
    usesheet.Range("F2").Value = initDc
    usesheet.Range("A1").Value = "Bridge"
    usesheet.Range("A2").Value = bridgename
    usesheet.Range("C1").Value = "Concrete Age(years)"
    usesheet.Range("C2").Value = servicelife

    usesheet.Range("A3").Value = "Depth(" & unit4x(unitset) & ")"
    usesheet.Range("B3").Value = "Cx,t measured(" & unit4cxt(unitset) & ")"
    usesheet.Range("C3").Value = "Cx,t predicted"
    usesheet.Range("D3").Value = "Difference square"

    usesheet.Range("A4").Value = x0
    usesheet.Range("B4").Value = c0
    usesheet.Cells(4, 3).Formula = "=B4*(1-erf((A4-A4)/(2*SQRT(F2*C2))))"
    usesheet.Cells(4, 4).Formula = "=(B4-C4)^2"

    'usesheet.Range("F1").Value = "Difussion Constant Least Square Estimate(" &
unit4dc(unitset) & ")"
    'usesheet.Range("F2").Value = initDc
```

```
    'usesheet.Range("A1").Value = "Co(" & unit4cxt(unitset) & ")"
    'usesheet.Range("A2").Value = c0
    'usesheet.Range("B1").Value = "at depth x0(" & unit4x(unitset) & ")"
    'usesheet.Range("B2").Value = x0
    'usesheet.Range("C1").Value = "Concrete Age"
    'usesheet.Range("C2").Value = servicelife
    'usesheet.Range("A3").Value = "Depth(" & unit4x(unitset) & ")"
    'usesheet.Range("B3").Value = "Cx,t measured(" & unit4cxt(unitset) & ")"
    'usesheet.Range("C3").Value = "Cx,t predicted"
    'usesheet.Range("D3").Value = "Difference square"

    Dim i As Long
    For i = 1 To datanum
        usesheet.Cells(i + 4, 1) = calx(i)
        usesheet.Cells(i + 4, 2) = calcxt(i)
        usesheet.Cells(i + 4, 3).Formula = "=B4*(1-erf((A" & i + 4 & "-
A4)/(2*SQRT(F2*C2))))"
        usesheet.Cells(i + 4, 4).Formula = "=(B" & i + 4 & "-C" & i + 4 & ")^2"
    Next i
    usesheet.Cells(datanum + 5, 4).Formula = "=SUM(D4:D" & datanum + 4 & ")"

    SolverOk SetCell:="$D$" & datanum + 5, MaxMinVal:=2, ByChange:="$F$2"
    SolverAdd CellRef:="$F$2", Relation:=3, FormulaText:="0.001"

    'Do not display the solver results dialog box;
    SolverSolve UserFinish:=True
    'Finish and keep the final results
    Solverfinish keepfinal:=1
    Dim Dcfinal As Double
    Dcfinal = usesheet.Range("F2").Value

    Dim currentsheet As String
    currentsheet = usesheet.Name
    'Dim totalsheets As Long
    'totalsheets = Sheets.Count

    Application.ScreenUpdating = True
    Dim xdata As Range, y1data As Range, y2data As Range
    Set xdata = Range(usesheet.Cells(4, 1), usesheet.Cells(datanum + 4, 1))
    Set y1data = Range(usesheet.Cells(4, 2), usesheet.Cells(datanum + 4, 2))
    Set y2data = Range(usesheet.Cells(4, 3), usesheet.Cells(datanum + 4, 3))

    Charts.Add
    ActiveChart.ChartType = xlXYScatter
    'ActiveChart.SetSourceData Source:=Sheets("Sheet4").Range("E10"),
PlotBy:=xlColumns
```

```
    'ActiveChart.SeriesCollection(1).Delete
    'ActiveChart.SeriesCollection.NewSeries
    ActiveChart.SeriesCollection.NewSeries
    ActiveChart.SeriesCollection(2).XValues = y1data
    ActiveChart.SeriesCollection(2).Values = xdata
    ActiveChart.SeriesCollection(2).Name = "=""Measurements"""
    ActiveChart.SeriesCollection(1).XValues = y2data
    ActiveChart.SeriesCollection(1).Values = xdata
    ActiveChart.SeriesCollection(1).Name = "=""Fitted Diffusion Curve"""
    ActiveChart.Location Where:=xlLocationAsObject, Name:=currentsheet
    With ActiveChart
        .HasTitle = False
        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Cx,t(" &
unit4cxt(unitset) & ")"
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Depth(" & unit4x(unitset)
& ")"
    End With
    'ActiveChart.ApplyDataLabels Type:=xlDataLabelsShowNone, LegendKey:=False
    ActiveChart.Axes(xlValue).Select
    With ActiveChart.Axes(xlValue)
        .MinimumScaleIsAuto = True
        .MaximumScaleIsAuto = True
        .MinorUnitIsAuto = True
        .MajorUnitIsAuto = True
        .Crosses = xlMaximum
        .ReversePlotOrder = True
        .ScaleType = xlLinear
        .DisplayUnit = xlNone
    End With
    ActiveChart.SeriesCollection(1).Select
    With Selection.Border
        .Weight = xlThin
        .LineStyle = xlAutomatic
    End With
    'With Selection
    '    .MarkerBackgroundColorIndex = xlAutomatic
    '    .MarkerForegroundColorIndex = xlAutomatic
    '    .MarkerStyle = xlAutomatic
    '    .Smooth = False
    '    .MarkerSize = 5
    '    .Shadow = False
    'End With
    'ActiveChart.ChartArea.Select
    With ActiveChart
```

```
        .HasTitle = True
        .ChartTitle.Characters.Text = bridgename & "'s chloride profile for " & servicelife &
" years: diffusion rate= " & Round(Dcfinal, 2) & "(mm^2/year), Co= " & Round(c0, 2) &
"(kg/m^3)"
    End With
    Application.CommandBars("Chart").Visible = False
    ActiveChart.ChartArea.Select
    Sheets(currentsheet).Name = bridgename

End With

End Sub

Private Sub Cmdhelp_Click()
    Call ShowHelp(400)
End Sub

Private Sub Label9_Click()

End Sub

Private Sub comboboxUnit_Change()
    If comboboxUnit.Value = "SI" Then
        unitset = 1
        Lblcxtunit.Caption = unit4cxtSI
        lblxunit.Caption = unit4xSI
        lblc0unit.Caption = unit4cxtSI
        lblx0unit.Caption = unit4xSI
        lbldc0unit.Caption = unit4dcSI
    ElseIf comboboxUnit.Value = "U.S. Customary" Then
        unitset = 2
        Lblcxtunit.Caption = unit4cxtUS
        lblxunit.Caption = unit4xUS
        lblc0unit.Caption = unit4cxtUS
        lblx0unit.Caption = unit4xUS
        lbldc0unit.Caption = unit4dcUS

    End If

End Sub


Private Sub UserForm_Initialize()

unit4dcSI = "mm^2/year"
unit4dcUS = "in^2/year"
```

```
unit4xSI = "mm"
unit4xUS = "in"
unit4cxtSI = "kg/m^3"
unit4cxtUS = "lb/in^3"

    'Add list entries to combo box. The value of each
    'entry matches the corresponding ListIndex value
    'in the combo box.
    comboboxUnit.AddItem "SI"        'ListIndex = 0
    comboboxUnit.AddItem "U.S. Customary"     'ListIndex = 1
    'Use drop-down list
    comboboxUnit.Style = fmStyleDropDownList
    comboboxUnit.Value = "SI"

        unitset = 1 'SI measurement unit
        Lblcxtunit.Caption = unit4cxtSI
        lblxunit.Caption = unit4xSI
        lblc0unit.Caption = unit4cxtSI
        lblx0unit.Caption = unit4xSI
        lbldc0unit.Caption = unit4dcSI

    comboboxUnit.Enabled = True

End Sub
```

Initiation Rate

```
Option Explicit
Public currentsheet1 As String


Private Sub CheckBox1_Change()
With Initrate
    If .CheckBox1.Value = True Then
        .combobox1.Enabled = True
        .OptionButton1.Enabled = True
        .OptionButton6.Enabled = True
        .OptionButton11.Enabled = True
        .OptionButton16.Enabled = True
        .TextBox1.Enabled = True
        .TextBox2.Enabled = True
        .TextBox11.Enabled = True
        .TextBox12.Enabled = True
        .RefEdit1.Enabled = True
        If .OptionButton1.Value = False And .OptionButton6.Value = False And _
            .OptionButton11.Value = False And .OptionButton16.Value = False Then
```

```
            .OptionButton1.Value = True
        End If
    ElseIf .CheckBox1.Value = False Then
        .combobox1.Enabled = False
        .OptionButton1.Enabled = False
        .OptionButton6.Enabled = False
        .OptionButton11.Enabled = False
        .OptionButton16.Enabled = False
        .TextBox1.Enabled = False
        .TextBox2.Enabled = False
        .TextBox11.Enabled = False
        .TextBox12.Enabled = False
        .RefEdit1.Enabled = False
   End If
End With
End Sub

Private Sub CheckBox2_Change()
With Initrate
    If CheckBox2.Value = True Then
        ComboBox2.Enabled = True
        OptionButton2.Enabled = True
        OptionButton7.Enabled = True
        OptionButton12.Enabled = True
        OptionButton17.Enabled = True
        TextBox4.Enabled = True
        TextBox3.Enabled = True
        TextBox14.Enabled = True
        TextBox13.Enabled = True
        RefEdit2.Enabled = True
        If .OptionButton2.Value = False And .OptionButton7.Value = False And _
            .OptionButton12.Value = False And .OptionButton17.Value = False Then

            .OptionButton2.Value = True
        End If
    ElseIf CheckBox2.Value = False Then
        ComboBox2.Enabled = False
        OptionButton2.Enabled = False
        OptionButton7.Enabled = False
        OptionButton12.Enabled = False
        OptionButton17.Enabled = False
        TextBox4.Enabled = False
        TextBox3.Enabled = False
        TextBox14.Enabled = False
        TextBox13.Enabled = False
        RefEdit2.Enabled = False
```

```
    End If
End With
End Sub

Private Sub CheckBox3_Change()
With Initrate
    If CheckBox3.Value = True Then
        ComboBox3.Enabled = True
        OptionButton3.Enabled = True
        OptionButton8.Enabled = True
        OptionButton13.Enabled = True
        OptionButton18.Enabled = True
        TextBox5.Enabled = True
        TextBox6.Enabled = True
        TextBox15.Enabled = True
        TextBox16.Enabled = True
        RefEdit3.Enabled = True
        If .OptionButton3.Value = False And .OptionButton8.Value = False And _
            .OptionButton13.Value = False And .OptionButton18.Value = False Then

            .OptionButton3.Value = True
        End If
    ElseIf CheckBox3.Value = False Then
        ComboBox3.Enabled = False
        OptionButton3.Enabled = False
        OptionButton8.Enabled = False
        OptionButton13.Enabled = False
        OptionButton18.Enabled = False
        TextBox5.Enabled = False
        TextBox6.Enabled = False
        TextBox15.Enabled = False
        TextBox16.Enabled = False
        RefEdit3.Enabled = False
    End If
End With
End Sub


Private Sub CheckBox4_Change()
With Initrate
    If CheckBox4.Value = True Then
        ComboBox4.Enabled = True
        OptionButton4.Enabled = True
        OptionButton9.Enabled = True
        OptionButton14.Enabled = True
        OptionButton19.Enabled = True
```

```
        TextBox7.Enabled = True
        TextBox8.Enabled = True
        TextBox17.Enabled = True
        TextBox18.Enabled = True
        RefEdit4.Enabled = True
        If .OptionButton4.Value = False And .OptionButton9.Value = False And _
           .OptionButton14.Value = False And .OptionButton19.Value = False Then

           .OptionButton4.Value = True
        End If
    ElseIf CheckBox4.Value = False Then
        ComboBox4.Enabled = False
        OptionButton4.Enabled = False
        OptionButton9.Enabled = False
        OptionButton14.Enabled = False
        OptionButton19.Enabled = False
        TextBox7.Enabled = False
        TextBox8.Enabled = False
        TextBox17.Enabled = False
        TextBox18.Enabled = False
        RefEdit4.Enabled = False
    End If
End With
End Sub


Private Sub CheckBox5_Change()
With Initrate
    If CheckBox5.Value = True Then
        ComboBox5.Enabled = True
        OptionButton5.Enabled = True
        OptionButton10.Enabled = True
        OptionButton15.Enabled = True
        OptionButton20.Enabled = True
        TextBox9.Enabled = True
        TextBox10.Enabled = True
        TextBox19.Enabled = True
        TextBox20.Enabled = True
        RefEdit5.Enabled = True
        If .OptionButton5.Value = False And .OptionButton10.Value = False And _
           .OptionButton15.Value = False And .OptionButton20.Value = False Then

           .OptionButton5.Value = True
        End If
    ElseIf CheckBox5.Value = False Then
        ComboBox5.Enabled = False
```

```
        OptionButton5.Enabled = False
        OptionButton10.Enabled = False
        OptionButton15.Enabled = False
        OptionButton20.Enabled = False
        TextBox9.Enabled = False
        TextBox10.Enabled = False
        TextBox19.Enabled = False
        TextBox20.Enabled = False
        RefEdit5.Enabled = False
    End If
End With
End Sub


Private Sub cmdback_Click()
    Unload Me
    Application.ScreenUpdating = True
    'Application.ScreenUpdating = False
    InputData.Show
End Sub

Private Sub cmdcancel_Click()
    Hide
End Sub

Private Sub Cmdhelp_Click()
    Call ShowHelp(300)

End Sub

Private Sub cmdsample_Click()
    currentsheet1 = ActiveSheet.Name
    Worksheets(currentsheet1).Activate

With Initrate
    Application.ScreenUpdating = True


    '*************************************************************
    '    Read initiation rates input                          *
    '*************************************************************
    'Define how many series shown on the chart, which is equal to the number of initiation
rates
    Dim numofrates As Long
    numofrates = 0
    If .CheckBox1.Value = True Then numofrates = numofrates + 1
```

```
    If .CheckBox2.Value = True Then numofrates = numofrates + 1
    If .CheckBox3.Value = True Then numofrates = numofrates + 1
    If .CheckBox4.Value = True Then numofrates = numofrates + 1
    If .CheckBox5.Value = True Then numofrates = numofrates + 1
    If numofrates < 1 Then
        MsgBox "Please input initiation rate!"
        'Show
        Exit Sub
    End If
    If numofrates > 5 Then
        MsgBox "Number of Initiation rates should be 1-5, please check!"
        'Show
        Exit Sub
    End If

    'initiation rate input
    Dim rateinputtype(5) As Long
    Dim ratelabel(5) As String
    Dim rateavg(5) As Double
    Dim rateminmax(5, 2) As Double
    Dim datacount(5) As Long
    Dim rateinputcolumn() As Double     'This array will be redimed after reading the input
cxtdata from the "initiation rate" interface.
    '
    'Chart series
    Dim seriesNew() As Series
    ReDim Preserve seriesNew(1 To 2 + 2 * numofrates) As Series
    'Estimated service years for different initiation rates
    Dim services() As Double    'For each initiation rate, there are 6 estimated service years
correponding to 2,4,6,12,20,30 cumulative percentiles.
    Dim quart() As Double, years() As Double
    ReDim Preserve services(1 To numofrates, 6) As Double
    ReDim Preserve quart(1 To numofrates, 1 To iterationtimes) As Double, years(1 To
numofrates, 1 To iterationtimes) As Double

    'define and redim the bootstrapped data
    Dim Codatasel() As Double, dcdatasel() As Double, xdatasel() As Double
    ReDim Preserve Codatasel(1 To iterationtimes) As Double
    ReDim Preserve dcdatasel(1 To iterationtimes) As Double
    ReDim Preserve xdatasel(1 To iterationtimes) As Double
    Dim cxtdatasel() As Double
    ReDim Preserve cxtdatasel(1 To numofrates, 1 To iterationtimes) As Double

    'Temporary variables defination
    'some temporary array, need to be redimmed if iterationtimes exceeds 500
    Dim z() As Double, znew() As Double, y() As Double, t() As Double
```

```
    ReDim Preserve z(1 To iterationtimes) As Double, znew(1 To iterationtimes) As
Double
    ReDim Preserve y(1 To iterationtimes) As Double, t(1 To iterationtimes) As Double
    'others
    Dim percentage As Double
    Dim ocell As Range, r As Long
    Dim Coselected As Long, dcselected As Long, xselected As Long
    Dim sum As Double
    Dim chservyears As Chart

    'Read initiation rates labels and parameters
    'Define default upper and lower parameters for triangular distribution cxt()
    Dim i As Long
    Dim tempmean As Double, tempsd As Double
    'ratedefault(1) = 1.2
    'ratedefault(2) = 2
    'ratedefault(3) = 3
    'ratedefault(4) = 4
    'ratedefault(5) = 5
    'Read rates label and min/max or data column from the excel spreadsheet.
    Dim s As Long
    Dim rngrate(5) As Range
    'For i = 1 To 5
    '    Set rngrate(i) = Range("")
    'Next i
    s = 0
    If .CheckBox1.Value = True Then
        s = s + 1
        If .combobox1.Value = "" Then
            MsgBox "Please define the label of the initiation rates!"
            'Show
            Exit Sub
        End If
        ratelabel(s) = .combobox1.Value
        If .OptionButton1.Value = True Then
            rateinputtype(s) = 1
        ElseIf .OptionButton6.Value = True Then
            rateinputtype(s) = 2
            rateminmax(s, 1) = Val(.TextBox1.Text)
            rateminmax(s, 2) = Val(.TextBox2.Text)
            'should check to rule out incorrect min/max parameters value.
            If rateminmax(s, 1) >= rateminmax(s, 2) Then
                MsgBox "Max value should be larger than Min value!"
                .TextBox2.SelStart = 0
                .TextBox2.SelLength = .TextBox2.TextLength
                'MsgBox .TextBox2.SelText
```

```
            .TextBox2.SetFocus
            Exit Sub
          End If
        ElseIf .OptionButton11.Value = True Then
          rateinputtype(s) = 3
          tempmean = Val(.TextBox11.Text)
          tempsd = Val(.TextBox12.Text)
          If tempsd <= 0 Then
            MsgBox "The standard deviation should be positive!"
            .TextBox12.SelStart = 0
            .TextBox12.SelLength = .TextBox12.TextLength
            'MsgBox .TextBox2.SelText
            .TextBox12.SetFocus
            Exit Sub
          End If
          rateminmax(s, 1) = tempmean - tempsd / 1.73205081
          rateminmax(s, 2) = tempmean + tempsd / 1.73205081
          'should check to rule out incorrect mean/sd parameters value.
          'also check if
          If rateminmax(s, 1) < 0 Then
            MsgBox "For the given mean/sd, the range of initiation rate includes negative
values!"
            .TextBox12.SelStart = 0
            .TextBox12.SelLength = .TextBox12.TextLength
            .TextBox12.SetFocus
           Exit Sub
          End If
        ElseIf .OptionButton16.Value = True Then
          rateinputtype(s) = 4
          If .RefEdit1.Value = "" Then
            MsgBox "Please choose the input data range!"
            '.RefEdit1.SetFocus
            Exit Sub
          End If
          Set rngrate(s) = Range(.RefEdit1.Value)
          'First check the data are from 3 columns
          If rngrate(s).Columns.Count <> 1 Then
            MsgBox "Check if the data are input from one column!"
            Exit Sub
          End If
        Else
          MsgBox "Choose one way to initialize the rate."
          'Show
          Exit Sub
        End If
    End If
```

```
    If .CheckBox2.Value = True Then
        s = s + 1
        If .ComboBox2.Value = "" Then
            MsgBox "Please define the label of the initiation rates!"
            Exit Sub
        End If
        ratelabel(s) = .ComboBox2.Value
        If .OptionButton2.Value = True Then
            rateinputtype(s) = 1
        ElseIf .OptionButton7.Value = True Then
            rateinputtype(s) = 2
            rateminmax(s, 1) = Val(.TextBox3.Text)
            rateminmax(s, 2) = Val(.TextBox4.Text)
            'should check to rule out incorrect min/max parameters value.
            If rateminmax(s, 1) >= rateminmax(s, 2) Then
                MsgBox "Max value should be larger than Min value!"
                .TextBox4.SelStart = 0
                .TextBox4.SelLength = .TextBox4.TextLength
                'MsgBox .TextBox2.SelText
                .TextBox4.SetFocus
                Exit Sub
            End If
        ElseIf .OptionButton12.Value = True Then
            rateinputtype(s) = 3
            tempmean = Val(.TextBox13.Text)
            tempsd = Val(.TextBox14.Text)
            If tempsd <= 0 Then
                MsgBox "The standard deviation should be positive!"
                .TextBox14.SelStart = 0
                .TextBox14.SelLength = .TextBox12.TextLength
                'MsgBox .TextBox2.SelText
                .TextBox14.SetFocus
                Exit Sub
            End If
            rateminmax(s, 1) = tempmean - tempsd / 1.73205081
            rateminmax(s, 2) = tempmean + tempsd / 1.73205081
            'should check to rule out incorrect mean/sd parameters value.
            If rateminmax(s, 1) < 0 Then
                MsgBox "For the given mean/sd, the range of initiation rate includes negative
values!"
                .TextBox14.SelStart = 0
                .TextBox14.SelLength = .TextBox12.TextLength
                .TextBox14.SetFocus
               Exit Sub
            End If
        ElseIf .OptionButton17.Value = True Then
```

```
        rateinputtype(s) = 4
        If .RefEdit2.Value = "" Then
            MsgBox "Please choose the input data range!"
            Exit Sub
        End If
        Set rngrate(s) = Range(.RefEdit2.Value)
        'First check the data are from 3 columns
        If rngrate(s).Columns.Count <> 1 Then
            MsgBox "Check if the data are input from one column!"
            Exit Sub
        End If
    Else
        MsgBox "Choose one way to initialize the rate."
        Exit Sub
    End If
End If
If .CheckBox3.Value = True Then
    s = s + 1
    If .ComboBox3.Value = "" Then
        MsgBox "Please define the label of the initiation rates!"
        Exit Sub
    End If
    ratelabel(s) = .ComboBox3.Value
    If .OptionButton3.Value = True Then
        rateinputtype(s) = 1
    ElseIf .OptionButton8.Value = True Then
        rateinputtype(s) = 2
        rateminmax(s, 1) = Val(.TextBox5.Text)
        rateminmax(s, 2) = Val(.TextBox6.Text)
        'should check to rule out incorrect min/max parameters value.
        If rateminmax(s, 1) >= rateminmax(s, 2) Then
            MsgBox "Max value should be larger than Min value!"
            .TextBox6.SelStart = 0
            .TextBox6.SelLength = .TextBox6.TextLength
            'MsgBox .TextBox2.SelText
            .TextBox6.SetFocus
            Exit Sub
        End If
    ElseIf .OptionButton13.Value = True Then
        rateinputtype(s) = 3
        tempmean = Val(.TextBox15.Text)
        tempsd = Val(.TextBox16.Text)
        If tempsd <= 0 Then
            MsgBox "The standard deviation should be positive!"
            .TextBox16.SelStart = 0
            .TextBox16.SelLength = .TextBox12.TextLength
```

```
            'MsgBox .TextBox2.SelText
            .TextBox16.SetFocus
            Exit Sub
          End If
          rateminmax(s, 1) = tempmean - tempsd / 1.73205081
          rateminmax(s, 2) = tempmean + tempsd / 1.73205081
          'should check to rule out incorrect mean/sd parameters value.
          If rateminmax(s, 1) < 0 Then
            MsgBox "For the given mean/sd, the range of initiation rate includes negative
values!"
            .TextBox16.SelStart = 0
            .TextBox16.SelLength = .TextBox12.TextLength
            .TextBox16.SetFocus
           Exit Sub
          End If
        ElseIf .OptionButton18.Value = True Then
          rateinputtype(s) = 4
          If .RefEdit3.Value = "" Then
            MsgBox "Please choose the input data range!"
            Exit Sub
          End If
          Set rngrate(s) = Range(.RefEdit3.Value)
          'First check the data are from 3 columns
          If rngrate(s).Columns.Count <> 1 Then
            MsgBox "Check if the data are input from one column!"
            Exit Sub
          End If
        Else
          MsgBox "Choose one way to initialize the rate."
          Exit Sub
        End If
      End If
      If .CheckBox4.Value = True Then
        s = s + 1
        If .ComboBox4.Value = "" Then
          MsgBox "Please define the label of the initiation rates!"
          Exit Sub
        End If
        ratelabel(s) = .ComboBox4.Value
        If .OptionButton4.Value = True Then
          rateinputtype(s) = 1
        ElseIf .OptionButton9.Value = True Then
          rateinputtype(s) = 2
          rateminmax(s, 1) = Val(.TextBox7.Text)
          rateminmax(s, 2) = Val(.TextBox8.Text)
          'should check to rule out incorrect min/max parameters value.
```

```
            If rateminmax(s, 1) >= rateminmax(s, 2) Then
                MsgBox "Max value should be larger than Min value!"
                .TextBox8.SelStart = 0
                .TextBox8.SelLength = .TextBox8.TextLength
                'MsgBox .TextBox2.SelText
                .TextBox8.SetFocus
                Exit Sub
            End If
        ElseIf .OptionButton14.Value = True Then
            rateinputtype(s) = 3
            tempmean = Val(.TextBox17.Text)
            tempsd = Val(.TextBox18.Text)
            If tempsd <= 0 Then
                MsgBox "The standard deviation should be positive!"
                .TextBox18.SelStart = 0
                .TextBox18.SelLength = .TextBox12.TextLength
                'MsgBox .TextBox2.SelText
                .TextBox18.SetFocus
                Exit Sub
            End If
            rateminmax(s, 1) = tempmean - tempsd / 1.73205081
            rateminmax(s, 2) = tempmean + tempsd / 1.73205081
            'should check to rule out incorrect mean/sd parameters value.
            If rateminmax(s, 1) < 0 Then
                MsgBox "For the given mean/sd, the range of initiation rate includes negative
values!"
                .TextBox18.SelStart = 0
                .TextBox18.SelLength = .TextBox12.TextLength
                .TextBox18.SetFocus
               Exit Sub
            End If
        ElseIf .OptionButton19.Value = True Then
            rateinputtype(s) = 4
            If .RefEdit4.Value = "" Then
                MsgBox "Please choose the input data range!"
                Exit Sub
            End If
            Set rngrate(s) = Range(.RefEdit4.Value)
            'First check the data are from 3 columns
            If rngrate(s).Columns.Count <> 1 Then
                MsgBox "Check if the data are input from one column!"
                Exit Sub
            End If
        Else
            MsgBox "Choose one way to initialize the rate."
            Exit Sub
```

```
        End If
    End If
    If .CheckBox5.Value = True Then
        s = s + 1
        If .ComboBox5.Value = "" Then
            MsgBox "Please define the label of the initiation rates!"
            Exit Sub
        End If
        ratelabel(s) = .ComboBox5.Value
        If .OptionButton5.Value = True Then
            rateinputtype(s) = 1
        ElseIf .OptionButton10.Value = True Then
            rateinputtype(s) = 2
            rateminmax(s, 1) = Val(.TextBox9.Text)
            rateminmax(s, 2) = Val(.TextBox10.Text)
            'should check to rule out incorrect min/max parameters value.
            If rateminmax(s, 1) >= rateminmax(s, 2) Then
                MsgBox "Max value should be larger than Min value!"
                .TextBox10.SelStart = 0
                .TextBox10.SelLength = .TextBox10.TextLength
                'MsgBox .TextBox2.SelText
                .TextBox10.SetFocus
                Exit Sub
            End If
        ElseIf .OptionButton15.Value = True Then
            rateinputtype(s) = 3
            tempmean = Val(.TextBox19.Text)
            tempsd = Val(.TextBox20.Text)
            If tempsd <= 0 Then
                MsgBox "The standard deviation should be positive!"
                .TextBox20.SelStart = 0
                .TextBox20.SelLength = .TextBox12.TextLength
                'MsgBox .TextBox2.SelText
                .TextBox20.SetFocus
                Exit Sub
            End If
            rateminmax(s, 1) = tempmean - tempsd / 1.73205081
            rateminmax(s, 2) = tempmean + tempsd / 1.73205081
            'should check to rule out incorrect mean/sd parameters value.
            If rateminmax(s, 1) < 0 Then
                MsgBox "For the given mean/sd, the range of initiation rate includes negative
values!"
                .TextBox20.SelStart = 0
                .TextBox20.SelLength = .TextBox12.TextLength
                .TextBox20.SetFocus
               Exit Sub
```

```
        End If
    ElseIf .OptionButton20.Value = True Then
        rateinputtype(s) = 4
        If .RefEdit5.Value = "" Then
            MsgBox "Please choose the input data range!"
            Exit Sub
        End If
        'First check the data are from 3 columns
        If rngrate(s).Columns.Count <> 1 Then
            MsgBox "Check if the data are input from one column!"
            Exit Sub
        End If
        Set rngrate(s) = Range(.RefEdit5.Value)
    Else
        MsgBox "Choose one way to initialize the rate."
        Exit Sub
    End If
End If
If s <> numofrates Then
    MsgBox "There is something wrong with the number of initiation rates!"
    numofrates = s
End If
For i = 1 To numofrates
    If rateinputtype(i) = 1 Then
        If ratelabel(i) = "Bare" Then
            rateminmax(i, 1) = 0.39
            rateminmax(i, 2) = 9.08
            rateavg(i) = 2.79
        ElseIf ratelabel(i) = "Calcium Nitrite" Then
            rateminmax(i, 1) = 7.5
            rateminmax(i, 2) = 16
            rateavg(i) = 12.5
        ElseIf ratelabel(i) = "Stainless Steel" Then
            rateminmax(i, 1) = 13
            rateminmax(i, 2) = 18.8
            rateavg(i) = 15.9
        ElseIf ratelabel(i) = "Galvannized" Then
            rateminmax(i, 1) = 0.97
            rateminmax(i, 2) = 22.7
            rateavg(i) = 6.97
        ElseIf ratelabel(i) = "Epoxy" Then
            rateminmax(i, 1) = 0.08
            rateminmax(i, 2) = 9
            rateavg(i) = 4.14
        End If
    End If
```

```
        'Input rate as sysmetric trianguluar distribution wth average=(min+max)/2
        If rateinputtype(i) = 2 Or rateinputtype(i) = 3 Then
            rateavg(i) = (rateminmax(i, 1) + rateminmax(i, 2)) / 2
        End If
    Next i
    '
    'rule out the save rate type and use the same default values
    Dim j As Long
    Dim temp As Long, temps As Long
    For i = 1 To numofrates
        For j = i + 1 To numofrates
            If ratelabel(i) = ratelabel(j) Then
                If (rateinputtype(i) = rateinputtype(j) And rateminmax(i, 1) = rateminmax(j, 1)
And _
                rateminmax(i, 2) = rateminmax(j, 2) And rateavg(i) = rateavg(j) And
rateinputtype(i) <> 4) Then
                    MsgBox "Rate " & ratelabel(i) & "is totally the same as rate " & ratelabel(j)
& ", please change or cancel one of them!"
                    Exit Sub
                End If
                If (rateinputtype(i) = 1 And rateinputtype(j) = 2 And rateminmax(i, 1) =
rateminmax(j, 1) And _
                rateminmax(i, 2) = rateminmax(j, 2) And rateavg(i) = rateavg(j)) Then
                    MsgBox "Rate " & ratelabel(i) & "is totally the same as rate " & ratelabel(j)
& ", please change or cancel one of them!"
                    Exit Sub
                End If
                If (rateinputtype(i) = 2 And rateinputtype(j) = 1 And rateminmax(i, 1) =
rateminmax(j, 1) And _
                rateminmax(i, 2) = rateminmax(j, 2) And rateavg(i) = rateavg(j)) Then
                    MsgBox "Rate " & ratelabel(i) & "is totally the same as rate " & ratelabel(j)
& ", please change or cancel one of them!"
                    Exit Sub
                End If
                If (rateinputtype(i) = rateinputtype(j) And rateinputtype(i) = 4) Then
                    If rngrate(i) = rngrate(j) Then
                        MsgBox "Rate " & ratelabel(i) & "is totally the same as rate " &
ratelabel(j) & ", please change or cancel one of them!"
                        Exit Sub
                    End If
                End If
            End If
            If ratelabel(i) <> ratelabel(j) Then
                If (rateinputtype(i) = rateinputtype(j) And rateminmax(i, 1) = rateminmax(j, 1)
And _
```

```
            rateminmax(i, 2) = rateminmax(j, 2) And rateavg(i) = rateavg(j) And
rateinputtype(i) <> 4) Then
                MsgBox "Initiation rate " & ratelabel(i) & " and the rate " & ratelabel(j) & "
are differen type but with the same parameters, this is wrong!"
                Exit Sub
            End If
            If (rateinputtype(i) = 1 And rateinputtype(j) = 2 And rateminmax(i, 1) =
rateminmax(j, 1) And _
            rateminmax(i, 2) = rateminmax(j, 2) And rateavg(i) = rateavg(j)) Then
                MsgBox "Initiation rate " & ratelabel(i) & " and the rate " & ratelabel(j) & "
are differen type but with the same parameters, this is wrong!"
                Exit Sub
            End If
            If (rateinputtype(i) = 2 And rateinputtype(j) = 1 And rateminmax(i, 1) =
rateminmax(j, 1) And _
            rateminmax(i, 2) = rateminmax(j, 2) And rateavg(i) = rateavg(j)) Then
                MsgBox "Initiation rate " & ratelabel(i) & " and the rate " & ratelabel(j) & "
are differen type but with the same parameters, this is wrong!"
                Exit Sub
            End If
            If (rateinputtype(i) = rateinputtype(j) And rateinputtype(i) = 4) Then
                If rngrate(i).Count = rngrate(j).Count Then
                    temps = 0
                    For temp = 1 To rngrate(i).Count
                        If rngrate(i).Cells(temp, 1).Value = rngrate(j).Cells(temp, 1).Value
Then temps = temps + 1
                    Next temp
                    If temps = rngrate(i).Count Then
                        MsgBox "Initiation rate " & ratelabel(i) & " and the rate " & ratelabel(j)
& " are differen type but with the same parameters, this is wrong!"
                        Exit Sub
                    End If
                End If
            End If
        End If
    Next j
Next i
'
'Read the input data range for correponding initiation rates.
'
Dim p As Long
Dim maxcount As Long
p = 0
For i = 1 To numofrates
    If rateinputtype(i) = 4 Then
        p = p + 1
```

```
            datacount(i) = rngrate(i).Count
            If p = 1 Then
               maxcount = datacount(i)
            Else
               If maxcount < datacount(i) Then maxcount = datacount(i)
            End If
         End If
      Next i
      'MsgBox "There are " & p & " initiation rates input from the spreadsheet."
      If p > 0 Then ReDim Preserve rateinputcolumn(1 To numofrates, 1 To maxcount) As
Double
      Dim check As Boolean
      For i = 1 To numofrates
         If rateinputtype(i) = 4 Then
            For r = 1 To datacount(i)
               Call checkstr(rngrate(i).Cells(r, 1).Value, check)
               If check = False Then
                  MsgBox "Please make sure all the data have numerical value!"
                  Exit Sub
               End If
               rateinputcolumn(i, r) = rngrate(i).Cells(r, 1).Value
               If rateinputcolumn(i, r) = 0 Then
                  MsgBox "There are some empty cells within input initiation rate data!"
                  Exit Sub
               End If
            Next r
         End If
      Next i

      Hide
      Application.ScreenUpdating = True
      Application.ScreenUpdating = False

      '
      '************************************************************
      '*              Start sampling!                             *
      '************************************************************
      '
      'go to sub function to get normal cumulative distribution table
      Call readnormtab
      'This command maybe should be moved to the beginning of the whole project.
      '

      '**********************************************************
      'sample dc,x,Co for iteration times.They are the same for different intitiation rates.
      'save sampled data
```

```
    For r = 1 To iterationtimes
        Randomize    ' Initialize random-number generator.
        Coselected = Int((Conum * Rnd) + 1)
        Randomize
        dcselected = Int((dcnum * Rnd) + 1)
        Randomize
        xselected = Int((xnum * Rnd) + 1)
        Codatasel(r) = Codata(Coselected)
        dcdatasel(r) = dcdata(dcselected)
        xdatasel(r) = xdata(xselected)
    Next r


    'sample Cxt from triangular distribution. save them in xdata array
    'which includes a sampling of uniform distribution
    Dim tempcxt() As Double
    Dim counttemp As Long, indextemp As Long
    ReDim Preserve tempcxt(1 To iterationtimes) As Double
    For r = 1 To numofrates
        If rateinputtype(r) = 1 Or rateinputtype(r) = 2 Or rateinputtype(r) = 3 Then
            'Generate the triangular distribution cxt if default or min/max/avg are selected
            Call triangular(tempcxt, iterationtimes, rateminmax(r, 2), rateminmax(r, 1),
rateavg(r))
        ElseIf rateinputtype(r) = 4 Then
            'The number of input initiation rates might be less than the iterationtimes,
            'What should we do?
            'Sample iterationtimes from the rateinputcolumn(1 to dataxount(i)) like co, dc and
x
            counttemp = datacount(r)
            For i = 1 To iterationtimes
                Randomize    ' Initialize random-number generator.
                indextemp = Int((counttemp * Rnd) + 1)
                tempcxt(i) = rateinputcolumn(r, indextemp)
            Next i
        End If
        For i = 1 To iterationtimes
            cxtdatasel(r, i) = tempcxt(i)
        Next i
    Next r
    '
    '**************************************************
    '*          Begin calculation                     *
    '**************************************************
    '
    For i = 1 To 6
        For j = 1 To 8
```

```
        rngout.Cells(j, i).Value = ""
    Next j
Next i

rngout.Cells(1, 1).Value = ""
rngout.Cells(1, 2).Value = "Estimated Service Life (years)"
rngout.Cells(2, 1).Value = "Percentile(%)"

'
Dim estimatepointindex(5) As Long
Dim inum As Long
For inum = 1 To numofrates
    For r = 1 To iterationtimes
        If Codatasel(r) = 0 Then
            MsgBox "The denominator (chosen Co) is 0, some error happens!"
            Show
            Exit Sub
        End If
        z(r) = 1 - cxtdatasel(inum, r) / Codatasel(r)
        If z(r) < 0 Then
            znew(r) = 0
        Else
            znew(r) = z(r)
        End If
        'given the probabilty, check normal distribution table to get the random variable x.
        y(r) = normtablook(znew(r))
        If y(r) = 0 Then
            t(r) = 9999
        Else
            If dcdatasel(r) = 0 Then
                MsgBox "The denominator (chosen Dc) is 0, some error happens!"
                Show
                Exit Sub
            End If
            t(r) = (xdatasel(r) / (2 * y(r))) ^ 2 / dcdatasel(r)
        End If
    Next r


    'Calculate the direct average years
    'sum = 0
    'For r = 1 To iterationtimes
    '    If t(r) <> 9999 Then sum = sum + t(r)
    'Next r
    'averaget = sum / iterationtimes
```

```
        '
        'Export the output in the output range
        'output labels
        'rngout.Cells(1, 2 + numofrates * 2 + 2 + 9 * (inum - 1)).Value = "Average of t"
        'rngout.Cells(1, 2 + numofrates * 2 + 3 + 9 * (inum - 1)).Value = "Cxt"
        'rngout.Cells(1, 2 + numofrates * 2 + 4 + 9 * (inum - 1)).Value = "Co"
        'rngout.Cells(1, 2 + numofrates * 2 + 5 + 9 * (inum - 1)).Value = "x"
        'rngout.Cells(1, 2 + numofrates * 2 + 6 + 9 * (inum - 1)).Value = "Dc"
        'rngout.Cells(1, 2 + numofrates * 2 + 7 + 9 * (inum - 1)).Value = "z"
        'rngout.Cells(1, 2 + numofrates * 2 + 8 + 9 * (inum - 1)).Value = "znew"
        'rngout.Cells(1, 2 + numofrates * 2 + 9 + 9 * (inum - 1)).Value = "y"
        'rngout.Cells(1, 2 + numofrates * 2 + 10 + 9 * (inum - 1)).Value = "t"
        'rngout.Cells(1, 14).Value = "rvX"
        'rngout.Cells(1, 15).Value = "cumulative prob"
        'rngout.Cells(1, 2 + (inum - 1) * 2 + 2).Value = "Years"
        'rngout.Cells(1, 2 + (inum - 1) * 2 + 1).Value = "quartile"
        'output values
        'rngout.Cells(2, 2 + numofrates * 2 + 2 + 9 * (inum - 1)).Value = averaget
        'For r = 1 To iterationtimes
        '    rngout.Cells(r + 1, 2 + numofrates * 2 + 3 + 9 * (inum - 1)).Value =
cxtdatasel(inum, r)
        '    rngout.Cells(r + 1, 2 + numofrates * 2 + 4 + 9 * (inum - 1)).Value = Codatasel(r)
        '    rngout.Cells(r + 1, 2 + numofrates * 2 + 5 + 9 * (inum - 1)).Value = xdatasel(r)
        '    rngout.Cells(r + 1, 2 + numofrates * 2 + 6 + 9 * (inum - 1)).Value = dcdatasel(r)
        '    rngout.Cells(r + 1, 2 + numofrates * 2 + 7 + 9 * (inum - 1)).Value = z(r)
        '    rngout.Cells(r + 1, 2 + numofrates * 2 + 8 + 9 * (inum - 1)).Value = znew(r)
        '    rngout.Cells(r + 1, 2 + numofrates * 2 + 9 + 9 * (inum - 1)).Value = y(r)
        '    rngout.Cells(r + 1, 2 + numofrates * 2 + 10 + 9 * (inum - 1)).Value = t(r)
        'Next r
        'output the normal table
        'For r = 1 To 301
        '    rngout.Cells(r + 1, 14).Value = erftab(r, 1)
        '    rngout.Cells(r + 1, 15).Value = erftab(r, 2)
        'Next r
        '
        '



        'sort the service year array t(r) to get years estimates distribution
        Call QuickSort(t, 1, iterationtimes)    'sort t(r) in ascending order, sorted array is
saved still in t(r)
        quart(inum, iterationtimes) = 1
        For r = iterationtimes - 1 To 1 Step -1
           If t(r) <> t(r + 1) Then
              quart(inum, r) = r / iterationtimes
           Else
```

```
            quart(inum, r) = quart(inum, r + 1) '
        End If
    Next r
    For r = 1 To iterationtimes
        years(inum, r) = t(r)
    Next r


    '
    'Calculate the service years estimates for bridge ,
    'save them in the services(p,r), p=1 to numofrates, r=1-6,correponding to
2,4,6,12,20.30 cumulative percentage
    '
    Dim percentages(6) As Long
    percentage = 0.12
    percentages(1) = 2
    percentages(2) = 4
    percentages(3) = 6
    percentages(4) = 12
    percentages(5) = 20
    percentages(6) = 30
    For r = 1 To 6
        rngout.Cells(2 + r, 1).Value = percentages(r)
    Next r


'*********************************************************************
    'SInce the large bootstrap times, use the approximated service years
    s = 1
    r = 1
    Do While s <= 6
        Do While quart(inum, r) < percentages(s) / 100
            r = r + 1   ' Increment Counter.
            If r = iterationtimes + 1 Then
                MsgBox "Percentiles are calculated incorrectly!"
                Show
                Exit Sub    ' Exit inner loop.
            End If
        Loop
        services(inum, s) = t(r)
        If s = 4 Then estimatepointindex(inum) = r
        s = s + 1
    Loop
```

```
'*************************************************************************
*
      'Dim diff1 As Double, diff2 As Double
      'percentage = 0.12   'Or input from the interface
      'Dim counter As Long
      '
      'r = 1
      'Do While quart(inum, r) < percentage
      '    r = r + 1   ' Increment Counter.
      '    If r = iterationtimes + 1 Then
      '        MsgBox "Quartiles are calculated incorrectly!"
      '        Exit Sub    ' Exit inner loop.
      '    End If
      'Loop
      'If quart(inum, r) = percentage Then
      '    services(inum) = t(r)
      'Else
      '    diff1 = t(r) - t(r - 1)
      '    diff2 = quart(inum, r) - quart(inum, r - 1)
      '    services(inum) = t(r - 1) + diff1 / diff2 * (percentage - quart(inum, r - 1))
      'End If
      '

'*************************************************************************
***

      'Output them into the worksheet
      rngout.Cells(2, 1 + inum).Value = ratelabel(inum)
      For r = 1 To 6
         rngout.Cells(2 + r, 1 + inum).Value = services(inum, r)
      Next r
   Next inum

   'Output the estimated service years curve on the new sheet2
   Dim rngout2 As Range
   Dim newsheet As Worksheet
   Set newsheet = Worksheets.Add
   'newsheet.Name = "ServiceLife-bridge " & bridgelabel

   newsheet.Activate
   Set rngout2 = Range("A1")
   'Range("A1:K10000").Clea
   rngout2.Cells(1, 1).Value = "Estimated Service Life Curve"
   'rngout2.Cells(1, 2).Value = ""
   'rngout2.Cells(1, 3).Value = ""
```

```
'rngout2.Cells(1, 4).Value = ""
For inum = 1 To numofrates
   rngout2.Cells(2, 2 * inum - 1).Value = ratelabel(inum)
Next inum
For inum = 1 To numofrates
   rngout2.Cells(3, 2 * inum - 1).Value = "Percentile"
   rngout2.Cells(3, 2 * inum).Value = "ServiceYears"
   For r = 1 To iterationtimes
      rngout2.Cells(3 + r, 2 * inum - 1).Value = quart(inum, r)
      rngout2.Cells(3 + r, inum * 2).Value = years(inum, r)
   Next r
Next inum
'Output the Cxt for each initiate rate
rngout2.Cells(1, 2 * numofrates + 2).Value = "Cxt"
For inum = 1 To numofrates
   rngout2.Cells(2, 2 * numofrates + 1 + inum).Value = ratelabel(inum)
Next inum
For inum = 1 To numofrates
   For r = 1 To iterationtimes
      rngout2.Cells(2 + r, 2 * numofrates + 1 + inum).Value = cxtdatasel(inum, r)
   Next r
Next inum

'
'Make the selection clear.
'Worksheets("sheet1").Activate
'rngout.Cells(1, 1).Select
'Selection.Clear
Application.ScreenUpdating = True
Application.ScreenUpdating = False
'ActiveSheet.Select
'
'************************************************************
'*  Create a chart sheet to show sample bootstap about the service life for bridge  *
'************************************************************
'
Dim ydatasource As Range, xdatasource As Range
'
'create a chart sheet
Charts.Add
With ActiveChart
   '.Visible = False
   'set chart type
   .ChartType = xlXYScatterSmoothNoMarkers
         '
```

```
        'add array t(r) as x, quartile as y, they are shown in the range object: ydatasource and
xdatasource
        '
        'Maybe there are several y-x, so define here numofrates to save the number of the
series
        'These series correspond to t(r, p) and quart(r,p), p=1 to numofrates, if
numofrates<>1, we have to define ydatasource(p) and xdatasource(p)
        'They share the same x and y axis, but have different series labels.
        'Dim yarray() As Double, xarray() As Double
        'ReDim Preserve yarray(1 To iterationtimes) As Double
        'ReDim Preserve xarray(1 To iterationtimes) As Double
        Dim markstyles(5) As String
        markstyles(1) = xlMarkerStyleSquare
        markstyles(2) = xlMarkerStyleTriangle
        markstyles(3) = xlMarkerStyleX
        markstyles(4) = xlMarkerStyleDot
        markstyles(5) = xlMarkerStyleDiamond
        'lineweight(1) = xlThick
        'lineweight(2) = xlMedium
        'lineweight(3) = xlThin
        'lineweight(4) = xlMedium
        'lineweight(5) = xlThin


        For r = 1 To numofrates
            '
            'Generate chart data source
            Set ydatasource = Range(rngout2.Cells(4, r * 2 - 1), rngout2.Cells(iterationtimes
+ 3, r * 2 - 1))
            Set xdatasource = Range(rngout2.Cells(4, r * 2), rngout2.Cells(iterationtimes + 3,
r * 2))
            'For i = 1 To iterationtimes
            '    yarray(i) = years(r, i)
            '    xarray(i) = quart(r, i)
            'Next i

            If .SeriesCollection.Count < r Then
               Set seriesNew(r) = .SeriesCollection.NewSeries
            Else
               Set seriesNew(r) = .SeriesCollection.item(r)
            End If
            With seriesNew(r)
               .Name = ratelabel(r)
               .Values = ydatasource
               .XValues = xdatasource
               '.Points(estimatepointindex(r)).MarkerStyle = markstyles(r)
```

```
        '.MarkerStyle = markstyles(r)
    End With
Next r
'
'Draw the quartile line to calibrate the service years estimates
Set seriesNew(numofrates + 1) = .SeriesCollection.NewSeries
With seriesNew(numofrates + 1)
        .Name = "12% percentile"
        .Values = Array(percentage, percentage, percentage)
        .XValues = Array(0, t(2), 100)
End With
'.Location Where:=xlLocationAsObject, Name:=currentsheet1

'For r = 1 To numofrates
'    Set seriesNew(numofrates + 1 + r) = .SeriesCollection.NewSeries
'    With seriesNew(numofrates + 1 + r)
'        .Values = Array(0, percentage / 2, percentage)
'        .XValues = Array(services(r), services(r), services(r))
'    End With
'Next r
'
'set chart axis properties
With .Axes(xlCategory)
    .HasTitle = True
    .AxisTitle.Caption = "Time to corrosion initiation(years)"
    .MinimumScale = 0
    .MaximumScale = 100
    .HasMajorGridlines = False
End With
With .Axes(xlValue)
    .HasTitle = True
    .AxisTitle.Caption = "Cumulative Percentile Initiated"
    .MinimumScale = 0
    .MaximumScale = 1
    .CrossesAt = 0
    .HasMajorGridlines = False
End With
'set the crossing point on the y axis
'.Axes(xlValue).CrossesAt = 0
'set no horizontal and vertical gridlines
'.Axes(xlValue).HasMajorGridlines = False
'.Axes(xlCategory).HasMajorGridlines = False
'
'set legend properties
'MsgBox .SeriesCollection.Count
.HasLegend = True
```

```
        'For r = 1 To numofrates
        '    .Legend.LegendEntries(numofrates + 1).Delete
        'Next r
        '
        'Format chart title
        .HasTitle = True
        With .ChartTitle
           .Caption = "Service Life Estimates for " & bridgelabel
           .Font.Size = 14
        End With
        '.Visible = True
        '.Select
        .ChartArea.Select
        .Deselect
    End With
    ActiveChart.Location Where:=xlLocationAsObject, Name:=currentsheet1
    'chservyears.ProtectChartObject = True

End With
Application.ScreenUpdating = True
End Sub


Private Sub combobox1_Change()
    If combobox1.MatchFound = False Then
        OptionButton1.Value = False
        OptionButton1.Enabled = False
    ElseIf combobox1.MatchFound = True Then
        'OptionButton1.Value = true
        OptionButton1.Enabled = True
    End If
End Sub

Private Sub combobox2_Change()
    If ComboBox2.MatchFound = False Then
        OptionButton2.Value = False
        OptionButton2.Enabled = False
    ElseIf ComboBox2.MatchFound = True Then
        'OptionButton1.Value = true
        OptionButton2.Enabled = True
    End If
End Sub

Private Sub combobox3_Change()
    If ComboBox3.MatchFound = False Then
        OptionButton3.Value = False
```

```
        OptionButton3.Enabled = False
    ElseIf ComboBox3.MatchFound = True Then
        'OptionButton1.Value = true
        OptionButton3.Enabled = True
    End If
End Sub

Private Sub combobox4_Change()
    If ComboBox4.MatchFound = False Then
        OptionButton4.Value = False
        OptionButton4.Enabled = False
    ElseIf ComboBox4.MatchFound = True Then
        'OptionButton1.Value = true
        OptionButton4.Enabled = True
    End If
End Sub

Private Sub combobox5_Change()
    If ComboBox5.MatchFound = False Then
        OptionButton5.Value = False
        OptionButton5.Enabled = False
    ElseIf ComboBox5.MatchFound = True Then
        'OptionButton1.Value = true
        OptionButton5.Enabled = True
    End If
End Sub


Private Sub OptionButton1_Change()
    With Initrate
        If .OptionButton1.Value = True Then
            .TextBox1.Enabled = False
            .TextBox2.Enabled = False
            .TextBox11.Enabled = False
            .TextBox12.Enabled = False
            .RefEdit1.Enabled = False
        ElseIf .OptionButton1.Value = False Then
            .TextBox1.Enabled = True
            .TextBox2.Enabled = True
            .TextBox11.Enabled = True
            .TextBox12.Enabled = True
            .RefEdit1.Enabled = True
        End If
    End With
End Sub
```

```
Private Sub OptionButton11_Change()
    With Initrate
        If .OptionButton11.Value = True Then
            .TextBox1.Enabled = False
            .TextBox2.Enabled = False
            .TextBox11.Enabled = True
            .TextBox12.Enabled = True
            .RefEdit1.Enabled = False
        ElseIf .OptionButton11.Value = False Then
            .TextBox1.Enabled = True
            .TextBox2.Enabled = True
            .TextBox11.Enabled = False
            .TextBox12.Enabled = False
            .RefEdit1.Enabled = True
        End If
    End With

End Sub


Private Sub OptionButton16_Change()
    With Initrate
        If .OptionButton16.Value = True Then
            .TextBox1.Enabled = False
            .TextBox2.Enabled = False
            .TextBox11.Enabled = False
            .TextBox12.Enabled = False
            .RefEdit1.Enabled = True
        ElseIf .OptionButton16.Value = False Then
            .TextBox1.Enabled = True
            .TextBox2.Enabled = True
            .TextBox11.Enabled = True
            .TextBox12.Enabled = True
            .RefEdit1.Enabled = False
        End If
    End With

End Sub


Private Sub OptionButton6_Change()
    With Initrate
        If .OptionButton6.Value = True Then
            .TextBox1.Enabled = True
            .TextBox2.Enabled = True
```

```
            .TextBox11.Enabled = False
            .TextBox12.Enabled = False
            .RefEdit1.Enabled = False
        ElseIf .OptionButton6.Value = False Then
            .TextBox1.Enabled = False
            .TextBox2.Enabled = False
            .TextBox11.Enabled = True
            .TextBox12.Enabled = True
            .RefEdit1.Enabled = True
        End If
    End With

End Sub


Private Sub OptionButton2_Change()
    With Initrate
        If .OptionButton2.Value = True Then
            .TextBox3.Enabled = False
            .TextBox4.Enabled = False
            .TextBox13.Enabled = False
            .TextBox14.Enabled = False
            .RefEdit2.Enabled = False
        ElseIf .OptionButton2.Value = False Then
            .TextBox3.Enabled = True
            .TextBox4.Enabled = True
            .TextBox13.Enabled = True
            .TextBox14.Enabled = True
            .RefEdit2.Enabled = True
        End If
    End With
End Sub


Private Sub OptionButton12_Change()
    With Initrate
        If .OptionButton12.Value = True Then
            .TextBox3.Enabled = False
            .TextBox4.Enabled = False
            .TextBox13.Enabled = True
            .TextBox14.Enabled = True
            .RefEdit2.Enabled = False
        ElseIf .OptionButton12.Value = False Then
            .TextBox3.Enabled = True
            .TextBox4.Enabled = True
            .TextBox13.Enabled = False
```

```
            .TextBox14.Enabled = False
            .RefEdit2.Enabled = True
        End If
    End With

End Sub


Private Sub OptionButton17_Change()
    With Initrate
        If .OptionButton17.Value = True Then
            .TextBox3.Enabled = False
            .TextBox4.Enabled = False
            .TextBox13.Enabled = False
            .TextBox14.Enabled = False
            .RefEdit2.Enabled = True
        ElseIf .OptionButton17.Value = False Then
            .TextBox3.Enabled = True
            .TextBox4.Enabled = True
            .TextBox13.Enabled = True
            .TextBox14.Enabled = True
            .RefEdit2.Enabled = False
        End If
    End With

End Sub


Private Sub OptionButton7_Change()
    With Initrate
        If .OptionButton7.Value = True Then
            .TextBox3.Enabled = True
            .TextBox4.Enabled = True
            .TextBox13.Enabled = False
            .TextBox14.Enabled = False
            .RefEdit2.Enabled = False
        ElseIf .OptionButton7.Value = False Then
            .TextBox3.Enabled = False
            .TextBox4.Enabled = False
            .TextBox13.Enabled = True
            .TextBox14.Enabled = True
            .RefEdit2.Enabled = True
        End If
    End With

End Sub
```

```
Private Sub OptionButton3_Change()
    With Initrate
        If .OptionButton3.Value = True Then
            .TextBox5.Enabled = False
            .TextBox6.Enabled = False
            .TextBox15.Enabled = False
            .TextBox16.Enabled = False
            .RefEdit3.Enabled = False
        ElseIf .OptionButton3.Value = False Then
            .TextBox5.Enabled = True
            .TextBox6.Enabled = True
            .TextBox15.Enabled = True
            .TextBox16.Enabled = True
            .RefEdit3.Enabled = True
        End If
    End With
End Sub

Private Sub OptionButton8_Change()
    With Initrate
        If .OptionButton8.Value = True Then
            .TextBox5.Enabled = True
            .TextBox6.Enabled = True
            .TextBox15.Enabled = False
            .TextBox16.Enabled = False
            .RefEdit3.Enabled = False
        ElseIf .OptionButton8.Value = False Then
            .TextBox5.Enabled = False
            .TextBox6.Enabled = False
            .TextBox15.Enabled = True
            .TextBox16.Enabled = True
            .RefEdit3.Enabled = True
        End If
    End With
End Sub

Private Sub OptionButton13_Change()
    With Initrate
        If .OptionButton13.Value = True Then
            .TextBox5.Enabled = False
            .TextBox6.Enabled = False
            .TextBox15.Enabled = True
            .TextBox16.Enabled = True
            .RefEdit3.Enabled = False
        ElseIf .OptionButton13.Value = False Then
```

```
            .TextBox5.Enabled = True
            .TextBox6.Enabled = True
            .TextBox15.Enabled = False
            .TextBox16.Enabled = False
            .RefEdit3.Enabled = True
        End If
    End With
End Sub

Private Sub OptionButton18_Change()
    With Initrate
        If .OptionButton18.Value = True Then
            .TextBox5.Enabled = False
            .TextBox6.Enabled = False
            .TextBox15.Enabled = False
            .TextBox16.Enabled = False
            .RefEdit3.Enabled = True
        ElseIf .OptionButton18.Value = False Then
            .TextBox5.Enabled = True
            .TextBox6.Enabled = True
            .TextBox15.Enabled = True
            .TextBox16.Enabled = True
            .RefEdit3.Enabled = False
        End If
    End With
End Sub

Private Sub OptionButton4_Change()
    With Initrate
        If .OptionButton4.Value = True Then
            .TextBox7.Enabled = False
            .TextBox8.Enabled = False
            .TextBox17.Enabled = False
            .TextBox18.Enabled = False
            .RefEdit4.Enabled = False
        ElseIf .OptionButton4.Value = False Then
            .TextBox7.Enabled = True
            .TextBox8.Enabled = True
            .TextBox17.Enabled = True
            .TextBox18.Enabled = True
            .RefEdit4.Enabled = True
        End If
    End With
End Sub

Private Sub OptionButton9_Change()
```

```
    With Initrate
        If .OptionButton9.Value = True Then
            .TextBox7.Enabled = True
            .TextBox8.Enabled = True
            .TextBox17.Enabled = False
            .TextBox18.Enabled = False
            .RefEdit4.Enabled = False
        ElseIf .OptionButton9.Value = False Then
            .TextBox7.Enabled = False
            .TextBox8.Enabled = False
            .TextBox17.Enabled = True
            .TextBox18.Enabled = True
            .RefEdit4.Enabled = True
        End If
    End With
End Sub

Private Sub OptionButton14_Change()
    With Initrate
        If .OptionButton14.Value = True Then
            .TextBox7.Enabled = False
            .TextBox8.Enabled = False
            .TextBox17.Enabled = True
            .TextBox18.Enabled = True
            .RefEdit4.Enabled = False
        ElseIf .OptionButton14.Value = False Then
            .TextBox7.Enabled = True
            .TextBox8.Enabled = True
            .TextBox17.Enabled = False
            .TextBox18.Enabled = False
            .RefEdit4.Enabled = True
        End If
    End With
End Sub

Private Sub OptionButton19_Change()
    With Initrate
        If .OptionButton19.Value = True Then
            .TextBox7.Enabled = False
            .TextBox8.Enabled = False
            .TextBox17.Enabled = False
            .TextBox18.Enabled = False
            .RefEdit4.Enabled = True
        ElseIf .OptionButton19.Value = False Then
            .TextBox7.Enabled = True
            .TextBox8.Enabled = True
```

```
            .TextBox17.Enabled = True
            .TextBox18.Enabled = True
            .RefEdit4.Enabled = False
        End If
    End With
End Sub

Private Sub OptionButton5_Change()
    With Initrate
        If .OptionButton5.Value = True Then
            .TextBox9.Enabled = False
            .TextBox10.Enabled = False
            .TextBox19.Enabled = False
            .TextBox20.Enabled = False
            .RefEdit5.Enabled = False
        ElseIf .OptionButton5.Value = False Then
            .TextBox9.Enabled = True
            .TextBox10.Enabled = True
            .TextBox19.Enabled = True
            .TextBox20.Enabled = True
            .RefEdit5.Enabled = True
        End If
    End With
End Sub

Private Sub OptionButton10_Change()
    With Initrate
        If .OptionButton10.Value = True Then
            .TextBox9.Enabled = True
            .TextBox10.Enabled = True
            .TextBox19.Enabled = False
            .TextBox20.Enabled = False
            .RefEdit5.Enabled = False
        ElseIf .OptionButton10.Value = False Then
            .TextBox9.Enabled = False
            .TextBox10.Enabled = False
            .TextBox19.Enabled = True
            .TextBox20.Enabled = True
            .RefEdit5.Enabled = True
        End If
    End With
End Sub

Private Sub OptionButton15_Change()
    With Initrate
        If .OptionButton15.Value = True Then
```

```
            .TextBox9.Enabled = False
            .TextBox10.Enabled = False
            .TextBox19.Enabled = True
            .TextBox20.Enabled = True
            .RefEdit5.Enabled = False
        ElseIf .OptionButton15.Value = False Then
            .TextBox9.Enabled = True
            .TextBox10.Enabled = True
            .TextBox19.Enabled = False
            .TextBox20.Enabled = False
            .RefEdit5.Enabled = True
        End If
    End With
End Sub

Private Sub OptionButton20_Change()
    With Initrate
        If .OptionButton20.Value = True Then
            .TextBox9.Enabled = False
            .TextBox10.Enabled = False
            .TextBox19.Enabled = False
            .TextBox20.Enabled = False
            .RefEdit5.Enabled = True
        ElseIf .OptionButton20.Value = False Then
            .TextBox9.Enabled = True
            .TextBox10.Enabled = True
            .TextBox19.Enabled = True
            .TextBox20.Enabled = True
            .RefEdit5.Enabled = False
        End If
    End With
End Sub


Private Sub UserForm_Initialize()

    If unitset = 1 Then
        Lblcxtunit.Caption = "unit: " & unit4cxtSI
    ElseIf unitset = 2 Then
        Lblcxtunit.Caption = "unit: " & unit4cxtUS
    End If


    'Add list entries to combo box. The value of each
    'entry matches the corresponding ListIndex value
    'in the combo box.
```

```
combobox1.AddItem "Bare"        'ListIndex = 0
combobox1.AddItem "Calcium Nitrite"     'ListIndex = 1
combobox1.AddItem "Stainless Steel"     'ListIndex = 2
combobox1.AddItem "Galvannized"       'ListIndex = 3
combobox1.AddItem "Epoxy"    'ListIndex = 4
combobox1.Value = "Bare"
'
ComboBox2.AddItem "Bare"    'ListIndex = 0
ComboBox2.AddItem "Calcium Nitrite"      'ListIndex = 1
ComboBox2.AddItem "Stainless Steel"    'ListIndex = 2
ComboBox2.AddItem "Galvannized"    'ListIndex = 3
ComboBox2.AddItem "Epoxy"      'ListIndex =4
ComboBox2.Value = "Calcium Nitrite"
'
ComboBox3.AddItem "Bare"    'ListIndex = 0
ComboBox3.AddItem "Calcium Nitrite"      'ListIndex = 1
ComboBox3.AddItem "Stainless Steel"    'ListIndex = 2
ComboBox3.AddItem "Galvannized"    'ListIndex = 3
ComboBox3.AddItem "Epoxy"      'ListIndex =4
ComboBox3.Value = "Stainless Steel"
'
ComboBox4.AddItem "Bare"    'ListIndex = 0
ComboBox4.AddItem "Calcium Nitrite"      'ListIndex = 1
ComboBox4.AddItem "Stainless Steel"    'ListIndex = 2
ComboBox4.AddItem "Galvannized"    'ListIndex = 3
ComboBox4.AddItem "Epoxy"      'ListIndex =4
ComboBox4.Value = "Galvannized"
'
ComboBox5.AddItem "Bare"    'ListIndex = 0
ComboBox5.AddItem "Calcium Nitrite"      'ListIndex = 1
ComboBox5.AddItem "Stainless Steel"    'ListIndex = 2
ComboBox5.AddItem "Galvannized"    'ListIndex = 3
ComboBox5.AddItem "Epoxy"      'ListIndex =4
ComboBox5.Value = "Epoxy"

'Use drop-down list
combobox1.Style = fmStyleDropDownCombo
ComboBox2.Style = fmStyleDropDownCombo
ComboBox3.Style = fmStyleDropDownCombo
ComboBox4.Style = fmStyleDropDownCombo
ComboBox5.Style = fmStyleDropDownCombo
'Combo box values are ListIndex values
'ComboBox1.BoundColumn = 0
'Set combo box to first entry
'combobox1.ListIndex = -1
'ComboBox2.ListIndex = -1
```

```
'ComboBox3.ListIndex = -1
'ComboBox4.ListIndex = -1
'ComboBox5.ListIndex = -1

combobox1.Enabled = False
ComboBox2.Enabled = False
ComboBox3.Enabled = False
ComboBox4.Enabled = False
ComboBox5.Enabled = False

OptionButton1.Enabled = False
OptionButton2.Enabled = False
OptionButton3.Enabled = False
OptionButton4.Enabled = False
OptionButton5.Enabled = False
OptionButton6.Enabled = False
OptionButton7.Enabled = False
OptionButton8.Enabled = False
OptionButton9.Enabled = False
OptionButton10.Enabled = False
OptionButton11.Enabled = False
OptionButton12.Enabled = False
OptionButton13.Enabled = False
OptionButton14.Enabled = False
OptionButton15.Enabled = False
OptionButton16.Enabled = False
OptionButton17.Enabled = False
OptionButton18.Enabled = False
OptionButton19.Enabled = False
OptionButton20.Enabled = False


TextBox1.Enabled = False
'TextBox1.Locked = True
TextBox2.Enabled = False
TextBox3.Enabled = False
TextBox4.Enabled = False
TextBox5.Enabled = False
TextBox6.Enabled = False
TextBox7.Enabled = False
TextBox8.Enabled = False
TextBox9.Enabled = False
TextBox10.Enabled = False
TextBox11.Enabled = False
TextBox12.Enabled = False
TextBox13.Enabled = False
```

```
    TextBox14.Enabled = False
    TextBox15.Enabled = False
    TextBox16.Enabled = False
    TextBox17.Enabled = False
    TextBox18.Enabled = False
    TextBox19.Enabled = False
    TextBox20.Enabled = False

    RefEdit1.Enabled = False
    RefEdit2.Enabled = False
    RefEdit3.Enabled = False
    RefEdit4.Enabled = False
    RefEdit5.Enabled = False

End Sub
```

Input Data

```
Option Explicit
Public CancelPressed As Boolean
'From first useform input, redefine them after input iterations


Private Sub cmdcancel_Click()
   ' Chart1.Delete
   CancelPressed = True
    Unload Me
End Sub

Private Sub Cmdhelp_Click()
    Call ShowHelp(300)
End Sub


Private Sub cmdnext_Click()
    'Hide
    'CancelPressed = False
With InputData
    '.Show
    Application.ScreenUpdating = True
    'Application.ScreenUpdating = False
    'if .CancelPressed Then Exit Sub



'*****************************************************************
**
```

```
    '*   Read Co, Dc, x data and bridge label from the 'inputdata' useform   *
'*************************************************************************
**
    '
    Dim rngCo As Range, rngx As Range, rngdc As Range
    If .txtbridge.Text = "" Then
        MsgBox "Please enter label of the bridge!"
        Exit Sub
    End If
    bridgelabel = .txtbridge.Text
    If .RefitCo.Value = "" Or .RefitDc.Value = "" Or .Refitx.Value = "" Then
        MsgBox "Please choose the input data range!"
        Exit Sub
    End If
    Set rngCo = Range(.RefitCo.Value)
    Set rngdc = Range(.RefitDc.Value)
    Set rngx = Range(.Refitx.Value)
    'First check the data are from 3 columns
    If rngCo.Columns.Count <> 1 Or rngdc.Columns.Count <> 1 Or rngx.Columns.Count
<> 1 Then
        MsgBox "Check if the data are input from three columns!"
        Exit Sub
    End If
    Conum = rngCo.Count
        ReDim Preserve Codata(1 To Conum) As Double
    dcnum = rngdc.Count
        ReDim Preserve dcdata(1 To dcnum) As Double
    xnum = rngx.Count
        ReDim Preserve xdata(1 To xnum) As Double
    '
    'Check if select the output range
    If .Refitout.Value = "" Then
        MsgBox "Please choose the output range!"
        Exit Sub
    End If
    Set rngout = Range(.Refitout.Value)
    'rngout.Cells(1, 1).Select
    '
    'Reas iterationtimes, if it exceeds 500, need redim data array.
    iterationtimes = Val(.txtiter.Text)
    If iterationtimes = 0 Then
        MsgBox "The iteration times should be greater than 0!"
        Exit Sub
    End If
    '
```

```
    '
    'Save the data from worksheet to the arrays Codata, dcdata and cxtdata
    Dim r As Long
    Dim check As Boolean
    For r = 1 To Conum
       Call checkstr(rngCo.Cells(r, 1).Value, check)
       If check = False Then
            MsgBox "Please make sure all the data have numerical value!"
            Exit Sub
       End If
       Codata(r) = rngCo.Cells(r, 1).Value
       If Codata(r) = 0 Then
            MsgBox "There are some empty cells within Co data, please check!"
            Exit Sub
       End If
   Next r
    For r = 1 To dcnum
       Call checkstr(rngdc.Cells(r, 1).Value, check)
       If check = False Then
            MsgBox "Please make sure all the data have numerical value!"
            Exit Sub
       End If
       dcdata(r) = rngdc.Cells(r, 1).Value
       If dcdata(r) = 0 Then
            MsgBox "There are some empty cells within Dc data, please check!"
            Exit Sub
       End If
    Next r
    For r = 1 To xnum
       Call checkstr(rngx.Cells(r, 1).Value, check)
       If check = False Then
            MsgBox "Please make sure all the data have numerical value!"
            Exit Sub
       End If
       xdata(r) = rngx.Cells(r, 1).Value
       If xdata(r) = 0 Then
            MsgBox "There are some empty cells within x data, please check!"
            Exit Sub
       End If
    Next r
    .Hide
End With

Initrate.Show
```

```
End Sub

Private Sub comboboxUnit_Change()

    If comboboxUnit.Value = "SI" Then
        unitset = 1
        lblxunit.Caption = unit4xSI
        lblc0unit.Caption = unit4cxtSI
        lbldcunit.Caption = unit4dcSI
    ElseIf comboboxUnit.Value = "U.S. Customary" Then
        unitset = 2
        lblxunit.Caption = unit4xUS
        lblc0unit.Caption = unit4cxtUS
        lbldcunit.Caption = unit4dcUS

    End If

End Sub

Private Sub RefitCo_Change()
    'Worksheets(currentsheet1).Activate
    Dim rngCo As Range
    If RefitCo.Value <> "" Then
        Set rngCo = Range(RefitCo.Value)
        lblconum.Caption = "Total " & Str(rngCo.Count) & " observations"
    End If
End Sub

Private Sub Refitx_Change()
    Dim rngx As Range
    If Refitx.Value <> "" Then
        Set rngx = Range(Refitx.Value)
        lblxnum.Caption = "Total " & Str(rngx.Count) & " observations"
    End If
End Sub


Private Sub RefitDc_Change()
    Dim rngdc As Range
    If RefitDc.Value <> "" Then
        Set rngdc = Range(RefitDc.Value)
        lbldcnum.Caption = "Total " & Str(rngdc.Count) & " observations"
    End If
End Sub

Private Sub UserForm_Initialize()
```

```
    unit4dcSI = "mm^2/year"
    unit4dcUS = "in^2/year"
    unit4xSI = "mm"
    unit4xUS = "in"
    unit4cxtSI = "kg/m^3"
    unit4cxtUS = "lb/in^3"
    'Add list entries to combo box. The value of each
    'entry matches the corresponding ListIndex value
    'in the combo box.
    comboboxUnit.AddItem "SI"        'ListIndex = 0
    comboboxUnit.AddItem "U.S. Customary"     'ListIndex = 1
    'Use drop-down list
    comboboxUnit.Style = fmStyleDropDownList
    comboboxUnit.Value = "SI"

        unitset = 1 'SI measurement unit
        lblxunit.Caption = unit4xSI
        lblc0unit.Caption = unit4cxtSI
        lbldcunit.Caption = unit4dcSI

    comboboxUnit.Enabled = True


End Sub

Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
CancelPressed = True
End Sub

Private Sub UserForm_Terminate()
CancelPressed = True
End Sub
```

ERF Function

```
Option Explicit

Public erftab(1 To 305, 1 To 2) As Double

'Public Declare Function SQRT Lib Math
Public Sub triangular(ByRef Data() As Double, ByVal itimes As Long, upper As Double,
lower As Double, avg As Double)
    Dim j As Long
    Dim x1 As Double, x2 As Double, z As Double, x As Double

    If avg = (lower + upper) / 2 Then
        'Symetric triangular ditribution
```

```
        For j = 1 To itimes
            Randomize    ' Initialize random-number generator.
            x1 = (upper - lower) * Rnd + lower
            Randomize    ' Initialize random-number generator.
            x2 = (upper - lower) * Rnd + lower
            z = (x1 + x2) / 2
            Data(j) = z
        Next j
    Else
        'Non-Symetric triangular ditribution
        For j = 1 To itimes
            Randomize
            x = Rnd
            If x < 0.5 Then
                z = 1.41421356 * Sqr(x) * (avg - lower) + lower
            Else
                z = upper - 1.41421356 * Sqr(1 - x) * (upper - avg)
            End If
            Data(j) = z
        Next j
    End If

End Sub

Public Function normtablook(ByVal prob As Double) As Double
    Dim x As Double
    'return the random variable x with the cumulative probability of prob in normal
distribution
    If prob = 0 Then
        x = 0
    ElseIf prob = 1 Then
        x = 3
    ElseIf prob > 0 And prob < 1 Then
        x = tablook(prob)
    Else
        MsgBox "Cumulative probability should be in the range [0,1]!"
        Exit Function
    End If
    normtablook = x

End Function

Public Function tablook(ByVal x As Double) As Double
    Dim counter As Long
    Dim diff1 As Double, diff2 As Double
```

```
    counter = 1
    Do While erftab(counter, 2) < x ' Inner loop.
        counter = counter + 1    ' Increment Counter.
        If counter = 302 Then    ' If condition is True.
            MsgBox "The cumulative probability of " & x & " can't be found from the table!"
' Set value of flag to False.
            Exit Function    ' Exit inner loop.
        End If
    Loop
    If erftab(counter, 2) = x Then
        tablook = erftab(counter, 1)
    Else
        diff2 = erftab(counter, 2) - erftab((counter - 1), 2)
        diff1 = erftab(counter, 1) - erftab((counter - 1), 1)
        tablook = erftab((counter - 1), 1) + diff1 / diff2 * (x - erftab((counter - 1), 2))
    End If

End Function

Public Sub readnormtab()
    Dim r As Long

    For r = 1 To 301
        erftab(r, 1) = (r - 1) / 100
    Next r
    'THe following are the cumulative probability of normal distribution
erftab(1, 2) = 0
erftab(2, 2) = 0.01128342
erftab(3, 2) = 0.02256458
erftab(4, 2) = 0.03384122
erftab(5, 2) = 0.04511111
erftab(6, 2) = 0.05637198
erftab(7, 2) = 0.06762159
erftab(8, 2) = 0.07885772
erftab(9, 2) = 0.09007813
erftab(10, 2) = 0.10128059
erftab(11, 2) = 0.11246292
erftab(12, 2) = 0.1236229
erftab(13, 2) = 0.13475835
erftab(14, 2) = 0.14586712
erftab(15, 2) = 0.15694703
erftab(16, 2) = 0.16799597
erftab(17, 2) = 0.17901181
erftab(18, 2) = 0.18999246
erftab(19, 2) = 0.20093584
erftab(20, 2) = 0.21183989
```

erftab(21, 2) = 0.22270259
erftab(22, 2) = 0.23352192
erftab(23, 2) = 0.24429591
erftab(24, 2) = 0.2550226
erftab(25, 2) = 0.26570006
erftab(26, 2) = 0.27632639
erftab(27, 2) = 0.28689972
erftab(28, 2) = 0.29741822
erftab(29, 2) = 0.30788007
erftab(30, 2) = 0.3182835
erftab(31, 2) = 0.32862676
erftab(32, 2) = 0.33890815
erftab(33, 2) = 0.349126
erftab(34, 2) = 0.35927866
erftab(35, 2) = 0.36936453
erftab(36, 2) = 0.37938205
erftab(37, 2) = 0.3893297
erftab(38, 2) = 0.39920598
erftab(39, 2) = 0.40900945
erftab(40, 2) = 0.4187387
erftab(41, 2) = 0.42839235
erftab(42, 2) = 0.43796909
erftab(43, 2) = 0.44746762
erftab(44, 2) = 0.45688669
erftab(45, 2) = 0.46622512
erftab(46, 2) = 0.47548172
erftab(47, 2) = 0.48465539
erftab(48, 2) = 0.49374505
erftab(49, 2) = 0.50274967
erftab(50, 2) = 0.51166826
erftab(51, 2) = 0.52049988
erftab(52, 2) = 0.52924362
erftab(53, 2) = 0.53789863
erftab(54, 2) = 0.54646409
erftab(55, 2) = 0.55493924
erftab(56, 2) = 0.56332336
erftab(57, 2) = 0.57161576
erftab(58, 2) = 0.57981581
erftab(59, 2) = 0.5879229
erftab(60, 2) = 0.5959365
erftab(61, 2) = 0.60385609
erftab(62, 2) = 0.61168122
erftab(63, 2) = 0.61941146
erftab(64, 2) = 0.62704644
erftab(65, 2) = 0.63458583
erftab(66, 2) = 0.64202932

erftab(67, 2) = 0.64937668
erftab(68, 2) = 0.6566277
erftab(69, 2) = 0.6637822
erftab(70, 2) = 0.67084005
erftab(71, 2) = 0.67780119
erftab(72, 2) = 0.68466526
erftab(73, 2) = 0.69143282
erftab(74, 2) = 0.6981037
erftab(75, 2) = 0.70467782
erftab(76, 2) = 0.71115543
erftab(77, 2) = 0.71753653
erftab(78, 2) = 0.72382144
erftab(79, 2) = 0.73001024
erftab(80, 2) = 0.73610324
erftab(81, 2) = 0.74210079
erftab(82, 2) = 0.74800314
erftab(83, 2) = 0.75381059
erftab(84, 2) = 0.75952362
erftab(85, 2) = 0.76514256
erftab(86, 2) = 0.77066793
erftab(87, 2) = 0.77610012
erftab(88, 2) = 0.78143972
erftab(89, 2) = 0.78668722
erftab(90, 2) = 0.79184313
erftab(91, 2) = 0.79690811
erftab(92, 2) = 0.80188274
erftab(93, 2) = 0.80676762
erftab(94, 2) = 0.81156347
erftab(95, 2) = 0.81627095
erftab(96, 2) = 0.82089072
erftab(97, 2) = 0.82542358
erftab(98, 2) = 0.82987023
erftab(99, 2) = 0.83423142
erftab(100, 2) = 0.838508
erftab(101, 2) = 0.84270074
erftab(102, 2) = 0.84681045
erftab(103, 2) = 0.85083795
erftab(104, 2) = 0.85478416
erftab(105, 2) = 0.8586499
erftab(106, 2) = 0.86243607
erftab(107, 2) = 0.86614353
erftab(108, 2) = 0.86977325
erftab(109, 2) = 0.87332612
erftab(110, 2) = 0.87680307
erftab(111, 2) = 0.88020504
erftab(112, 2) = 0.88353297

erftab(113, 2) = 0.88678785
erftab(114, 2) = 0.88997064
erftab(115, 2) = 0.8930823
erftab(116, 2) = 0.89612382
erftab(117, 2) = 0.89909617
erftab(118, 2) = 0.90200037
erftab(119, 2) = 0.9048374
erftab(120, 2) = 0.90760826
erftab(121, 2) = 0.91031396
erftab(122, 2) = 0.91295549
erftab(123, 2) = 0.91553386
erftab(124, 2) = 0.91805008
erftab(125, 2) = 0.92050516
erftab(126, 2) = 0.92290011
erftab(127, 2) = 0.92523593
erftab(128, 2) = 0.92751362
erftab(129, 2) = 0.92973418
erftab(130, 2) = 0.93189862
erftab(131, 2) = 0.93400793
erftab(132, 2) = 0.93606311
erftab(133, 2) = 0.93806514
erftab(134, 2) = 0.94001502
erftab(135, 2) = 0.94191371
erftab(136, 2) = 0.94376219
erftab(137, 2) = 0.94556143
erftab(138, 2) = 0.94731239
erftab(139, 2) = 0.94901602
erftab(140, 2) = 0.95067329
erftab(141, 2) = 0.95228511
erftab(142, 2) = 0.95385243
erftab(143, 2) = 0.95537617
erftab(144, 2) = 0.95685725
erftab(145, 2) = 0.95829656
erftab(146, 2) = 0.95969502
erftab(147, 2) = 0.96105351
erftab(148, 2) = 0.96237289
erftab(149, 2) = 0.96365406
erftab(150, 2) = 0.96489786
erftab(151, 2) = 0.96610514
erftab(152, 2) = 0.96727674
erftab(153, 2) = 0.96841349
erftab(154, 2) = 0.96951621
erftab(155, 2) = 0.97058569
erftab(156, 2) = 0.97162273
erftab(157, 2) = 0.97262812
erftab(158, 2) = 0.97360263

erftab(159, 2) = 0.97454701
erftab(160, 2) = 0.97546201
erftab(161, 2) = 0.97634838
erftab(162, 2) = 0.97720683
erftab(163, 2) = 0.97803809
erftab(164, 2) = 0.97884284
erftab(165, 2) = 0.97962178
erftab(166, 2) = 0.98037558
erftab(167, 2) = 0.98110492
erftab(168, 2) = 0.98181044
erftab(169, 2) = 0.98249279
erftab(170, 2) = 0.98315259
erftab(171, 2) = 0.98379046
erftab(172, 2) = 0.98440701
erftab(173, 2) = 0.98500283
erftab(174, 2) = 0.9855785
erftab(175, 2) = 0.98613459
erftab(176, 2) = 0.98667167
erftab(177, 2) = 0.98719027
erftab(178, 2) = 0.98769094
erftab(179, 2) = 0.9881742
erftab(180, 2) = 0.98864055
erftab(181, 2) = 0.9890905
erftab(182, 2) = 0.98952454
erftab(183, 2) = 0.98994316
erftab(184, 2) = 0.9903468
erftab(185, 2) = 0.99073595
erftab(186, 2) = 0.99111103
erftab(187, 2) = 0.99147249
erftab(188, 2) = 0.99182075
erftab(189, 2) = 0.99215622
erftab(190, 2) = 0.99247932
erftab(191, 2) = 0.99279043
erftab(192, 2) = 0.99308994
erftab(193, 2) = 0.99337822
erftab(194, 2) = 0.99365565
erftab(195, 2) = 0.99392257
erftab(196, 2) = 0.99417933
erftab(197, 2) = 0.99442628
erftab(198, 2) = 0.99466372
erftab(199, 2) = 0.994892
erftab(200, 2) = 0.99511141
erftab(201, 2) = 0.99532226
erftab(202, 2) = 0.99552485
erftab(203, 2) = 0.99571945
erftab(204, 2) = 0.99590635

erftab(205, 2) = 0.99608581
erftab(206, 2) = 0.9962581
erftab(207, 2) = 0.99642346
erftab(208, 2) = 0.99658215
erftab(209, 2) = 0.99673441
erftab(210, 2) = 0.99688046
erftab(211, 2) = 0.99702053
erftab(212, 2) = 0.99715484
erftab(213, 2) = 0.99728361
erftab(214, 2) = 0.99740702
erftab(215, 2) = 0.99752529
erftab(216, 2) = 0.99763861
erftab(217, 2) = 0.99774715
erftab(218, 2) = 0.99785111
erftab(219, 2) = 0.99795065
erftab(220, 2) = 0.99804594
erftab(221, 2) = 0.99813715
erftab(222, 2) = 0.99822444
erftab(223, 2) = 0.99830795
erftab(224, 2) = 0.99838783
erftab(225, 2) = 0.99846423
erftab(226, 2) = 0.99853728
erftab(227, 2) = 0.99860712
erftab(228, 2) = 0.99867387
erftab(229, 2) = 0.99873766
erftab(230, 2) = 0.99879861
erftab(231, 2) = 0.99885682
erftab(232, 2) = 0.99891242
erftab(233, 2) = 0.99896551
erftab(234, 2) = 0.9990162
erftab(235, 2) = 0.99906457
erftab(236, 2) = 0.99911073
erftab(237, 2) = 0.99915478
erftab(238, 2) = 0.99919679
erftab(239, 2) = 0.99923686
erftab(240, 2) = 0.99927506
erftab(241, 2) = 0.99931149
erftab(242, 2) = 0.9993462
erftab(243, 2) = 0.99937928
erftab(244, 2) = 0.9994108
erftab(245, 2) = 0.99944083
erftab(246, 2) = 0.99946942
erftab(247, 2) = 0.99949665
erftab(248, 2) = 0.99952257
erftab(249, 2) = 0.99954724
erftab(250, 2) = 0.99957071

erftab(251, 2) = 0.99959305
erftab(252, 2) = 0.9996143
erftab(253, 2) = 0.9996345
erftab(254, 2) = 0.99965371
erftab(255, 2) = 0.99967198
erftab(256, 2) = 0.99968934
erftab(257, 2) = 0.99970584
erftab(258, 2) = 0.99972151
erftab(259, 2) = 0.9997364
erftab(260, 2) = 0.99975054
erftab(261, 2) = 0.99976397
erftab(262, 2) = 0.99977671
erftab(263, 2) = 0.99978881
erftab(264, 2) = 0.99980029
erftab(265, 2) = 0.99981118
erftab(266, 2) = 0.99982151
erftab(267, 2) = 0.99983131
erftab(268, 2) = 0.9998406
erftab(269, 2) = 0.99984941
erftab(270, 2) = 0.99985776
erftab(271, 2) = 0.99986567
erftab(272, 2) = 0.99987316
erftab(273, 2) = 0.99988026
erftab(274, 2) = 0.99988698
erftab(275, 2) = 0.99989335
erftab(276, 2) = 0.99989938
erftab(277, 2) = 0.99990508
erftab(278, 2) = 0.99991048
erftab(279, 2) = 0.99991559
erftab(280, 2) = 0.99992042
erftab(281, 2) = 0.99992499
erftab(282, 2) = 0.99992931
erftab(283, 2) = 0.99993339
erftab(284, 2) = 0.99993725
erftab(285, 2) = 0.9999409
erftab(286, 2) = 0.99994434
erftab(287, 2) = 0.9999476
erftab(288, 2) = 0.99995067
erftab(289, 2) = 0.99995358
erftab(290, 2) = 0.99995632
erftab(291, 2) = 0.9999589
erftab(292, 2) = 0.99996134
erftab(293, 2) = 0.99996364
erftab(294, 2) = 0.99996582
erftab(295, 2) = 0.99996787
erftab(296, 2) = 0.9999698

```
erftab(297, 2) = 0.99997162
erftab(298, 2) = 0.99997333
erftab(299, 2) = 0.99997495
erftab(300, 2) = 0.99997647
erftab(301, 2) = 0.99997791

End Sub
```

Menu Code

```
Global Const ItemName = "Bridge Corrosion Analysis"
Sub CreateCustomMenu()
  With Application.CommandBars("Worksheet Menu
Bar").Controls.Add(Type:=msoControlPopup, Temporary:=True)
      .Caption = "&Bridge Corrosion Analysis"
      .BeginGroup = True
      .Visible = True
      With .Controls.Add(Before:=1)
         .OnAction = "callsample"
         .Caption = "&Service Life Estimates"
         .Visible = True
      End With
      With .Controls.Add(Before:=2)
         .OnAction = "callcaldc"
         .Caption = "&Determine Diffusion Constant(Dc)"
         .Visible = True
      End With
      With .Controls.Add(Before:=3)
         .OnAction = "callhelp"
         .Caption = "&Help"
         .Visible = True
      End With
      With .Controls.Add(Before:=4)
         .OnAction = "callAbout"
         .Caption = "&About Bridge Corrosion"
         .Visible = True
      End With
   End With

 End Sub


Sub DeleteCustomMenu()
 For Each Control In Application.CommandBars.ActiveMenuBar.Controls
   If (Control.Caption = ItemName) Or (Control.Caption = "&Bridge Corrosion
Analysis") Then Control.Delete
 Next
```

```
End Sub
```

Module 1

```
Public bridgelabel As String
Public iterationtimes As Long
Public Codata() As Double, dcdata() As Double, xdata() As Double
Public Conum As Long, xnum As Long, dcnum As Long
Public rngout As Range

Public unitset As String
Public unit4cxtSI As String
Public unit4cxtUS As String
Public unit4xSI As String
Public unit4xUS As String
Public unit4dcSI As String
Public unit4dcUS As String



Private Sub callsample()
    '
    InputData.Show

End Sub
Private Sub callhelp()
    '
    Call ShowHelp(0)
End Sub
Private Sub callAbout()
    '
   aboutform.Show
End Sub

Public Sub callcaldc()
    '
    dccalc.Show
End Sub

Public Sub ShowHelp(ByVal ID As Long)
    If ID = 0 Then
        Application.Help ThisWorkbook.Path & "\bridge.hlp"
    Else
        Application.Help ThisWorkbook.Path & "\bridge.hlp", ID
    End If

End Sub
```

Open User Form

```
Sub OpenInputData()
  'With InputData
  '   .Show
  'End With
End Sub
```

Sort

```
Option Explicit

Public Sub checkstr(ByVal str1, Optional ByRef trueornot As Boolean)
    Dim i As Long, j As Long, s As Long
    Dim str2(1 To 11) As String * 1
    Dim str3 As String * 1

    For i = 0 To 9
        str2(i + 1) = i
    Next i
    str2(11) = "."

    s = 0
    For i = 1 To Len(str1)
        str3 = Mid(str1, i, 1)
        For j = 1 To 11
            If str3 = str2(j) Then GoTo ext1
        Next j
        GoTo ext2
ext1:  s = s + 1
    Next i
ext2: If s = Len(str1) Then
        trueornot = True
    Else
        trueornot = False
    End If

End Sub

' ****************************************************************
' This subroutine takes an array as an argument, and will sort the
' array in ascending ASCII value.

' Example: Call QuickSort iArray ' The program will sort iArray
' Supply a lngLeft and/or lngRight value if you only want to sort
' a portion of the array.
' Example: Call QuickSort iArray, 2, 5 ' The program sorts elements
```

```
' iArray(2), iArray(3), iArray(4), and iArray(5), leaving iArray(0),
' iArray(1), and iArray(6-up) in their original order.

' By: Craig H. Rettig
' Date: July 2002

' Get more free code samples at http://www.BitbucketHeaven.com/
' ************************************************************

Public Sub QuickSort(ByRef vntArr() As Double, _
   Optional ByVal lngLeft As Long = -2, _
   Optional ByVal lngRight As Long = -2)

   Dim i, j, lngMid As Long
   Dim vntTestVal As Variant

   If lngLeft = -2 Then lngLeft = LBound(vntArr)
   If lngRight = -2 Then lngRight = UBound(vntArr)

   If lngLeft < lngRight Then
      lngMid = (lngLeft + lngRight) \ 2
      vntTestVal = vntArr(lngMid)
      i = lngLeft
      j = lngRight
      Do
         Do While vntArr(i) < vntTestVal
            i = i + 1
         Loop
         Do While vntArr(j) > vntTestVal
            j = j - 1
         Loop
         If i <= j Then
            Call SwapElements(vntArr, i, j)
            i = i + 1
            j = j - 1
         End If
      Loop Until i > j

      ' Optimize sort by sorting smaller segment first
      If j <= lngMid Then
         Call QuickSort(vntArr, lngLeft, j)
         Call QuickSort(vntArr, i, lngRight)
      Else
         Call QuickSort(vntArr, i, lngRight)
         Call QuickSort(vntArr, lngLeft, j)
      End If
```

```
    End If
End Sub


' Used in QuickSort function
Private Sub SwapElements(ByRef vntItems As Variant, _
    ByVal lngItem1 As Long, _
    ByVal lngItem2 As Long)

    Dim vntTemp As Variant

    vntTemp = vntItems(lngItem2)
    vntItems(lngItem2) = vntItems(lngItem1)
    vntItems(lngItem1) = vntTemp
End Sub
```