ภาคผนวก

ในภาคผนวกนี้จะรวบรวม CODE ของทุกโปรแกรมย่อย โดยจะเรียงรายชื่อตามลำดับอักษรของชื่อโปรแกรม

1. การประกาศตัวแปรทั่วไปที่ใช้ในทุกโปรแกรมย่อย

```
Option Explicit
Option Base 1
Private M As Integer, NR As Integer, NRJ As Integer, E As Single, EL() As Single
Private ZI() As Single, JRL() As Single, NLJ As Integer, Uy() As Single, NTL As Integer
Private AJ() As Single, AML() As Single, ND As Integer, N As Integer, NDJ As Single
Private ID() As Integer, LML() As Integer, NJ As Integer, J0 As Integer, LN As Integer
Private DJ() As Single, AM() As Single, AR() As Single, P As Single, A As Single, _
        WU AsSingle
Private TYP As Integer, WT As Single, MMT As Single, NPL As Integer, NUL As Integer
Private NML As Integer, D As Single, RTT() As Single
Const MD = 4
Const NB = 4
```

2. โปรแกรมย่อย BANFAC

```
Private Sub BANFAC(N As Integer, NB As Integer, A() As Single)
        Dim J As Integer, J1 As Integer, J2 As Integer, SUM As Single
        Dim I As Integer, K As Integer, I1 As Integer, TEMP As Single
        If A(1, 1) <= 0 Then
                MsgBox "Stiffness matrix is singular for continuous beam", vbOKOnly
                Exit Sub
        End If
        For J = 2 To N
                J1 = J - 1
                J2 = J - NB + 1
                If J2 < 1 Then J2 = 1
                If J1 <> 1 Then
                        For I = 2 To J1
                                I1 = I - 1
                                If I1 >= J2 Then
                                        SUM = A(I, J - I + 1)
                                        For K = J2 To I1
                                                SUM = SUM - A(K, I - K + 1) * A(K, J - K + 1)
                                        Next K
                                        A(I, J - I + 1) = SUM
                                End If
                        Next I
                End If
                SUM = A(J, 1)
                For K = J2 To J1
                        TEMP = A(K, J - K + 1) / A(K, 1)
                        SUM = SUM - TEMP * A(K, J - K + 1)
                        A(K, J - K + 1) = TEMP
                Next K
```

```
            If SUM <= 0 Then
                  MsgBox "Stiffness matrix is singular for continuous beam", vbOKOnly
                  Exit Sub
            End If
            A(J, 1) = SUM
      Next J
End Sub
```

3. โปรแกรมย่อย BANSOL

```
Private Sub BANSOL(N As Integer, NB As Integer, U() As Single, B() As Single, X() As Single)
        Dim I As Integer, J As Integer, K1 As Integer, K As Integer, K2 As Integer, I1 As Integer
        Dim SUM As Single
        For I = 1 To N
                J = I - NB + 1
                If I <= NB Then J = 1
                SUM = B(I)
                K1 = I - 1
                If J <= K1 Then
                        For K = J To K1
                                SUM = SUM - U(K, I - K + 1) * X(K)
                        Next K
                End If
                X(I) = SUM
        Next I
        For I = 1 To N
                X(I) = X(I) / U(I, 1)
        Next I
        For I1 = 1 To N
                I = N - I1 + 1
                J = I + NB - 1
                If J > N Then J = N
                SUM = X(I)
                K2 = I + 1
                If K2 <= J Then
                        For K = K2 To J
                                SUM = SUM - U(I, K - I + 1) * X(K)
                        Next K
                End If
        X(I) = SUM
    Next I1
End Sub
```

4. โปรแกรมย่อย cmdJointRe

```
Private Sub cmdJointRe_Click()
    Dim J As Integer, K As Integer, N1 As Integer
    On Error GoTo Err_Hand
    ReDim JRL(ND)
    ReDim ID(ND)
    imgRes.Visible = True
    For J = 1 To ND
        JRL(J) = 0
    Next J
    List1.AddItem "==================="
    List1.AddItem "Joint  Restraints"
    List1.AddItem "==================="
    List1.AddItem "Joint    JR1    JR2"
    For J = 1 To NRJ
        K = InputBox("Input  joint  index", "Joint  Restraints", "Joint", 6174.1, 1420)
        JRL(2 * K - 1) = InputBox("Input  JR1", "Joint  Restraints", "JR1", 6174.1, 1420)
        JRL(2 * K) = InputBox("Input  JR2", "Joint  Restraints", "JR2", 6174.1, 1420)
        List1.AddItem K & "        " & JRL(2 * K - 1) & "        " & JRL(2 * K)
    Next J
    N1 = 0
    For J = 1 To ND
        N1 = N1 + JRL(J)
        If JRL(J) > 0 Then
            D(J) = N + N1
        Else
            ID(J) = J - N1
        End If
    Next J
    cmdLoadD.Enabled = True
```

```
        imgRes.Visible = False
        Exit Sub
        Err_Hand:
                MsgBox "Please input 'Joint Restraint' again", vbOKOnly, "ERROR DATA"
End Sub
```

5. โปรแกรมย่อย cmdLoadD

```
Private Sub cmdLoadD_Click()
        Dim J As Integer, K As Integer, I As Integer, FEM(8) As Single
        On Error GoTo Err_Hand
        cmdPoint.Enabled = True
        cmdUnif.Enabled = True
        cmdTriL.Enabled = True
        cmdMMT.Enabled = True
        LN = LN + 1
        List1.AddItem "=========="
        List1.AddItem "Load  Data"
        List1.AddItem "=========="
        List1.AddItem "Loading  No." & LN
        NLJ = InputBox("Input  number  of  loaded  joints", "Load  Data", "NLJ", 6174.1, 1420)
        NDJ = InputBox("Input  number  of  displacement  joints", "Load  Data", "NDJ", 6174.1, _
                1420)
        NPL = InputBox("Input  number  of  point  load", "Load  Data", "NPL", 6174.1, 1420)
        NUL = InputBox("Input  number  of  uniform  load", "Load  Data", "NUL", 6174.1, 1420)
        NTL = InputBox("Input  number  of  triangelar  load", "Load  Data", "NTL", 6174.1, 1420)
        NML = InputBox("Input  number  of  moment  load", "Load  Data", "NML", 6174.1, 1420)
        List1.AddItem "Number  of  loaded  joints  =  " & NLJ
        List1.AddItem "Number  of  displacement  joints  =  " & NDJ
        List1.AddItem "Number  of  point  load  =  " & NPL
        List1.AddItem "Number  of  uniform  load  =  " & NUL
        List1.AddItem "Number  of  triangelar  load  =  " & NTL
        List1.AddItem "Number  of  moment  load  =  " & NML
        ReDim AJ(ND)
        ReDim Uy(NJ)
        ReDim RTT(NJ)
        For J = 1 To ND
        AJ(J) = 0
Next J
For J = 1 To NJ
        Uy(J) = 0
        RTT(J) = 0
Next J
If NLJ <> 0 Then
        imgJL.Visible = True
        List1.AddItem "================="
        List1.AddItem "Actions  At  Joints"
        List1.AddItem "================="
        List1.AddItem "Joint    AJ1    AJ2"
        For J = 1 To NLJ
                K = InputBox("Input  joint  index", "Action  At  Joints", "JOINT", _
                    6174.1, _1420)
                AJ(2 * K - 1) = InputBox("Input  AJ1", "Action At  Joints", "AJ1", _
                              6174.1, 1420)
                AJ(2 * K) = InputBox("Input  AJ2", "Action  At  Joints", "AJ2", _
                           6174.1, 1420)
                List1.AddItem K & "        " & AJ(2 * K - 1) & "        " & AJ(2 * K)
        Next J
End If
ReDim AML(4, M)
ReDim LML(M)
For J = 1 To M
        LML(J) = 0
        For K = 1 To 4
                AML(K, J) = 0
        Next K
Next J
```

```
imgJL.Visible = False
If NDJ <> 0 Then
    imgDJ.Visible = True
    List1.AddItem "=============================="
    List1.AddItem "Displacement  At  Joints"
    List1.AddItem "=============================="
    List1.AddItem "Joint     Uy     Rotation"
    For I = 1 To NDJ
        K = InputBox("Input  joint  index", "Displacement  At  Joints", _
            "JOINT",  6174.1, 1420)
        Uy(K) = InputBox("Input  displacement  at  joints  " & K & " , " &  _
                "Uy" , "Displacement  At  Joints", "Uy", 6174.1, 1420)
        RTT(K) = InputBox("Input  rotation  at joints  " & K, _
                 "Rotation  At  Joint",  "RTT", 6174.1, 1420)
        List1.AddItem K & "     " & Uy(K) & "     " & RTT(K)
        FEM(1) = 12 * E * ZI(K - 1) * Uy(K) / EL(K - 1) ^ 3
        FEM(2) = 6 * E * ZI(K - 1) * Uy(K) / EL(K - 1) ^ 2
        FEM(3) = 12 * E * ZI(K) * Uy(K) / EL(K) ^ 3
        FEM(4) = 6 * E * ZI(K) * Uy(K) / EL(K) ^ 2
        FEM(5) = 6 * E * ZI(K - 1) * RTT(K) / EL(K - 1) ^ 2
        FEM(6) = 2 * E * ZI(K - 1) * RTT(K) / EL(K - 1)
        FEM(7) = 6 * E * ZI(K) * RTT(K) / EL(K) ^ 2
        FEM(8) = 2 * E * ZI(K) * RTT(K) / EL(K)
        Select Case K
            Case 1
                AML(1, K) = FEM(3) + FEM(7)
                AML(2, K) = FEM(4) + 2 * FEM(8)
                AML(3, K) = -FEM(3) + FEM(7)
                AML(4, K) = FEM(4) + FEM(8)
                LML(K) = 1
```

```
                    Case NJ
                            AML(1, K - 1) = -FEM(1) + FEM(5)
                            AML(2, K - 1) = -FEM(2) + FEM(6)
                            AML(3, K - 1) = FEM(1) + FEM(5)
                            AML(4, K - 1) = -FEM(2) + 2 * FEM(6)
                            LML(K - 1) = 1
                    Case Else
                            AML(1, K - 1) = -FEM(1) + FEM(5)
                            AML(2, K - 1) = -FEM(2) + FEM(6)
                            AML(3, K - 1) = FEM(1) + FEM(5)
                            AML(4, K - 1) = -FEM(2) + 2 * FEM(6)
                            LML(K - 1) = 1
                            AML(1, K) = FEM(3) + FEM(7)
                            AML(2, K) = FEM(4) + 2 * FEM(8)
                            AML(3, K) = -FEM(3) + FEM(7)
                            AML(4, K) = FEM(4) + FEM(8)
                            LML(K) = 1
            End Select
        Next I
End If
imgDJ.Visible = False
If NPL <> 0 Or NUL <> 0 Or NTL <> 0 Or NML <> 0 Then
        Image1.Visible = True
        cmdPoint.Visible = True
        cmdUnif.Visible = True
        cmdTriL.Visible = True
        cmdMMT.Visible = True
End If
If NPL = 0 Then cmdPoint.Enabled = False
If NUL = 0 Then cmdUnif.Enabled = False
```

```
        If NTL = 0 Then cmdTriL.Enabled = False
        If NML = 0 Then cmdMMT.Enabled = False
        If NPL = 0 And NUL = 0 And NTL = 0 And NML = 0 Then cmdResult.Enabled = True
        Exit Sub
Err_Hand:
        MsgBox "Please input 'Load Data' again", vbOKOnly, "ERROR DATA"
End Sub
```

6. โปรแกรมย่อย cmdMemIn

```
Private Sub cmdMemIn_Click()
        Dim J As Integer, I As Integer
        On Error GoTo Err_Hand
        ReDim EL(M)
        ReDim ZI(M)
        List1.AddItem "============================"
        List1.AddItem "Member  Information"
        List1.AddItem "============================"
        List1.AddItem "Member    Length    Moment of inertia"
        For J = 1 To M
                I = InputBox("Input member index", "Member Information", _
                    "MEMBER", 6174.1, 1420)
                EL(I) = InputBox("Input length of member" & I, "Member Information", _
                          "EL", 6174.1, _ 1420)
                ZI(I) = InputBox("Input moment of inertia" & I, "Member Information", "ZI",
                6174.1, _ 1420)
                List1.AddItem I & "       " & EL(I) & "       " & ZI(I)
        Next J
        cmdJointRe.Enabled = True
        Exit Sub
Err_Hand:
        MsgBox "Please input 'Member Information' again", vbOKOnly, "ERROR DATA"
End Sub
```

7. โปรแกรมย่อย cmdMMT

```
Private Sub cmdMMT_Click()
        Dim I As Integer, K As Integer
        On Error GoTo Err_Hand
        imgMMT.Visible = True
        List1.AddItem "============"
        List1.AddItem "Moment  Load"
        List1.AddItem "============"
        List1.AddItem "Member    MMT    d"
        For K = 1 To NML
                I = InputBox("Input  member  number", "Moment  Load", "MEMBER", _
                    6174.1, 1420)
                MMT = InputBox("Input  value  of  moment , MMT", "Moment  Load", _
                        "MMT", 6174.1, 1420)
                D = InputBox("Input  distance , d", "Moment  Load", "d", 6174.1, 1420)
                List1.AddItem I & "     " & MMT & "     " & D
                AML(1, I) = AML(1, I) + 6 * MMT * D * (EL(I) - D) / EL(I) ^ 2
                AML(2, I) = AML(2, I) + MMT * (EL(I) - D) / EL(I) * (3 * D / EL(I) - 1)
                AML(3, I) = AML(3, I) + (-6 * MMT * D * (EL(I) - D) / EL(I) ^ 2)
                AML(4, I) = AML(4, I) + MMT * D / EL(I) * (3 * (EL(I) - D) / EL(I) - 1)
                LML(I) = 1
        Next K
        cmdMMT.Enabled = False
        cmdResult.Enabled = True
        imgMMT.Visible = False
        Exit Sub
Err_Hand:
        MsgBox "Please  input  'Load  Data'  again", vbOKOnly, "ERROR  DATA"
        imgMMT.Visible = False
        Image1.Visible = False
        cmdPoint.Visible = False
        cmdUnif.Visible = False
        cmdTriL.Visible = False
        cmdMMT.Visible = False
    End Sub
```

8. โปรแกรมย่อย cmdPoint

```
Private Sub cmdPoint_Click()
        Dim K As Integer, I As Integer
        On Error GoTo Err_Hand
        imgPoint.Visible = True
        List1.AddItem "=============="
        List1.AddItem "Point  Load"
        List1.AddItem "=============="
        List1.AddItem "Member     P     a"
        For K = 1 To NPL
                I = InputBox("Input  member  index", "Point  Load", "MEMBER", 6174.1, 1420)
                P = InputBox("Input  value  of  point  load , P", "Point Load", "P", 6174.1, 1420)
                A = InputBox("Input  distance , a", "Point  Load", "a", 6174.1, 1420)
                List1.AddItem I & "      " & P & "      " & A
                AML(1, I) = AML(1, I) + (-P * (EL(I) - A) / EL(I) * (A * (EL(I) - A) / _
                                EL(I) ^ 2 - A ^ 2 / EL(I) ^ 2 + 1))
                AML(2, I) = AML(2, I) + (-P * A * (EL(I) - A) ^ 2 / EL(I) ^ 2)
                AML(3, I) = AML(3, I) + (-P * A / EL(I) * (A * (EL(I) - A) /  _
                                EL(I) ^ 2 - (EL(I) - A) ^ 2 / EL(I) ^ 2 + 1))
                AML(4, I) = AML(4, I) + (P * A ^ 2 * (EL(I) - A) / EL(I) ^ 2)
                LML(I) = 1
        Next K
        cmdPoint.Enabled = False
        cmdResult.Enabled = True
        imgPoint.Visible = False
        Exit Sub
Err_Hand:
        MsgBox "Please  input  'Load  Data'  again", vbOKOnly, "ERROR  DATA"
        imgPoint.Visible = False
        Image1.Visible = False
        cmdPoint.Visible = False
        cmdUnif.Visible = False
        cmdTriL.Visible = False
        cmdMMT.Visible = False
End Sub
```

9. โปรแกรมย่อย cmdPrint

```
Private Sub cmdPrint_Click()
        Dim I As Integer
        On Error GoTo Err_Hand
        Printer.Print "================"
        Printer.Print "Structural  Data"
        Printer.Print "================"
        Printer.Print "Number  Of  Members  =  " & M
        Printer.Print "Number  Of  Support  Restraints  =  " & NR
        Printer.Print "Number  Of  Restrainted  Joints  =  " & NRJ
        Printer.Print "Modulus  Of  Elasticity  =  " & E
        Printer.Print "Number  Of  Joints  =  " & NJ
        Printer.Print "Number  Of  Degree  Of  Freedom  =  " & N
        Printer.Print "======================"
        Printer.Print "Member  Information"
        Printer.Print "======================"
        Printer.Print "Member    Length    Moment Of Inertia"
        For I = 1 To M
                Printer.Print I & "    " & EL(I) & "    " & ZI(I)
        Next I
        Printer.Print "Member    JR1    JR2"
        For I = 1 To NJ
                Printer.Print I & "    " & JRL(2 * I - 1) & "    " & JRL(2 * I)
        Next I
        Printer.Print "==========="
        Printer.Print "Load  Data"
        Printer.Print "==========="
        Printer.Print "Loading  No.  " & LN
        Printer.Print "Number  Of  Loaded  Joints = " & NLJ
        Printer.Print "Number  Of  Displacement  Joints = " & NDJ
Printer.Print "Number  Of  Point  Loads = " & NPL
Printer.Print "Number  Of  Uniform  Loads = " & NUL
Printer.Print "Number  Of  Trianfular  Loads = " & NTL
Printer.Print "Number  Of  Moment  Loads = " & NML
If NLJ <> 0 Then
        Printer.Print "=========================="
        Printer.Print "Actions  At  Joints"
        Printer.Print "=========================="
        Printer.Print "Joint    AJ1    AJ2"
        For I = 1 To NJ
                Printer.Print I & "    " & AJ(2 * I - 1) & "    " & AJ(2 * I)
        Next I
End If
If NDJ <> 0 Then
        Printer.Print "=============================="
        Printer.Print "Displacement  At  Joints"
        Printer.Print "=============================="
        Printer.Print "Joint    Uy    Rotation"
        For I = 1 To NJ
                Printer.Print I & "    " & Uy(I) & RTT(I)
        Next I
        Printer.Print ""
End If
Printer.Print "======================="
Printer.Print "Joint  Displacement"
Printer.Print "======================="
Printer.Print "Joint    DJ1    DJ2"
For I = 1 To NJ
        Printer.Print I & "    " & DJ(2 * I - 1) & "    " & DJ(2 * I)
Next I
```

```
        Printer.Print "=========================="
        Printer.Print "Member  End - Actions"
        Printer.Print "=========================="
        Printer.Print "Member    AM1    AM2    AM3    AM4"
        For I = 1 To M
            Printer.Print I & "    " & AM(1, I) & "    " & AM(2, I) & "    " & AM(3, I) & _
            "    " & AM(4, I)
        Next I
        Printer.Print "====================="
        Printer.Print "Support  Reactions"
        Printer.Print "====================="
        Printer.Print "Joint    AR1    AR2"
        For I = 1 To NJ
            Printer.Print I & "    " & AR(2 * I - 1) & "    " & AR(2 * I)
        Next I
            Printer.EndDoc
        Exit Sub
Err_Hand:
        MsgBox "Please  input  data  again", vbOKOnly, "ERROR  DATA"
End Sub
```

10. โปรแกรมย่อย cmdQUIT

```
Private Sub cmdQUIT_Click()
        frmBeam.Hide
        frmMain.Show
End Sub
```

11. โปรแกรมย่อย cmdResult

```
Private Sub cmdResult_Click()
        Dim J As Integer, K As Integer, SCM1 As Single, SCM2 As Single
        Dim SCM3 As Single, SFF() As Single, I1 As Integer, I2 As Integer
        Dim AE() As Single, JR As Integer, IR As Integer, DF() As Single
        Dim AC() As Single, IC As Integer, ITEM As Integer, IM(4) As Integer
        Dim AMD() As Single, DM() As Single, I As Integer, J1 As Integer
        Dim J2 As Integer, N1 As Integer, JE As Integer, SM(4, 4) As Single
        On Error GoTo Err_Hand
        Image1.Visible = False
        cmdPoint.Visible = False
        cmdUnif.Visible = False
        cmdTriL.Visible = False
        cmdMMT.Visible = False
        ReDim SFF(N, NB)
        ReDim AE(ND)
        ReDim AR(ND)
        For J = 1 To ND
                AE(J) = 0
                AR(J) = 0
        Next J
        For J = 1 To N
                For K = 1 To NB
                        SFF(J, K) = 0
                Next K
        Next J
        For I = 1 To M
                SCM1 = 4 * E * ZI(I) / EL(I)
                SCM2 = 1.5 * SCM1 / EL(I)
                SCM3 = 2 * SCM2 / EL(I)
```

```
SM(1, 1) = SCM3
SM(1, 2) = SCM2
SM(1, 3) = -SCM3
SM(1, 4) = SCM2
SM(2, 2) = SCM1
SM(2, 3) = -SCM2
SM(2, 4) = SCM1 / 2
SM(3, 3) = SCM3
SM(3, 4) = -SCM2
SM(4, 4) = SCM1
IM(1) = 2 * I - 1
IM(2) = 2 * I
IM(3) = 2 * I + 1
IM(4) = 2 * I + 2
For J = 1 To MD
        I1 = IM(J)
        If JRL(I1) <= 0 Then
                For K = J To MD
                        I2 = IM(K)
                        If JRL(I2) <= 0 Then
                                IR = ID(I1)
                                IC = ID(I2)
                                If IR >= IC Then
                                        ITEM = IR
                                        IR = IC
                                        IC = ITEM
                                End If
                                IC = IC - IR + 1
                                SFF(IR, IC) = SFF(IR, IC) + SM(J, K)
                        End If
```

```
            Next K
        End If
    Next J
Next I
Call BANFAC(N, NB, SFF())
If NDJ <> 0 Or NPL <> 0 Or NUL <> 0 Or NTL <> 0 Or NML <> 0 Then
    For I = 1 To M
        If LML(I) <> 0 Then
            AE(2 * I - 1) = AE(2 * I - 1) - AML(1, I)
            AE(2 * I) = AE(2 * I) - AML(2, I)
            AE(2 * I + 1) = AE(2 * I + 1) - AML(3, I)
            AE(2 * I + 2) = AE(2 * I + 2) - AML(4, I)
        End If
    Next I
End If
ReDim AC(ND)
For J = 1 To ND
    JR = ID(J)
    AC(JR) = AJ(J) + AE(J)
Next J
ReDim DF(N)
ReDim DJ(ND)
Call BANSOL(N, NB, SFF(), AC(), DF())
    J = N + 1
For K = 1 To ND
    JE = ND - K + 1
    If JRL(JE) = 0 Then
        J = J - 1
        DJ(JE) = DF(J)
    Else
```

```
            DJ(JE) = 0
        End If
    Next K
    List1.AddItem "=========================="
    List1.AddItem "Joint  Displacement"
    List1.AddItem "=========================="
    List1.AddItem "Joint    DJ1    DJ2"
    For J = 1 To NJ
        List1.AddItem J & "    " & DJ(2 * J - 1) & "    " & DJ(2 * J)
    Next J
    List1.AddItem "=========================="
    List1.AddItem "Member  End - Actions"
    List1.AddItem "=========================="
    List1.AddItem "Member    AM1    AM2    AM3    AM4"
    ReDim AM(MD, M)
    For I = 1 To M
        SCM1 = 4 * E * ZI(I) / EL(I)
        SCM2 = 1.5 * SCM1 / EL(I)
        SCM3 = 2 * SCM2 / EL(I)
        SM(1, 1) = SCM3
        SM(1, 2) = SCM2
        SM(1, 3) = -SCM3
        SM(1, 4) = SCM2
        SM(2, 2) = SCM1
        SM(2, 3) = -SCM2
        SM(2, 4) = SCM1 / 2
        SM(3, 3) = SCM3
        SM(3, 4) = -SCM2
        SM(4, 4) = SCM1
        IM(1) = 2 * I - 1
```

```
IM(2) = 2 * I
IM(3) = 2 * I + 1
IM(4) = 2 * I + 2
For J = 2 To 4
        J0 = J - 1
        For K = 1 To J0
                SM(J, K) = SM(K, J)
        Next K
Next J
ReDim AMD(MD)
ReDim DM(MD)
For J = 1 To MD
        AMD(J) = 0
        For K = 1 To MD
                I1 = IM(K)
                DM(K) = DJ(I1)
                AMD(J) = AMD(J) + SM(J, K) * DM(K)
        Next K
        AM(J, I) = AML(J, I) + AMD(J)
        I2 = IM(J)
        If JRL(I2) <> 0 Then
                AR(I2) = AR(I2) + AMD(J)
        End If
Next J
AM(1, I) = Round(AM(1, I), 4)
AM(2, I) = Round(AM(2, I), 4)
AM(3, I) = Round(AM(3, I), 4)
AM(4, I) = Round(AM(4, I), 4)
List1.AddItem I & "    " & AM(1, I) & "    " & AM(2, I) & "    " & AM(3, I) & _
                "    " & AM(4, I)
```

```
    Next I
    For J = 1 To ND
        If JRL(J) <> 0 Then
            AR(J) = AR(J) - AJ(J) - AE(J)
        End If
    Next J
    List1.AddItem "=================="
    List1.AddItem "Support  Reactions"
    List1.AddItem "=================="
    List1.AddItem "Joint     AR1     AR2"
    For J = 1 To NJ
        J1 = 2 * J - 1
        J2 = 2 * J
        N1 = JRL(J1) + JRL(J2)
        If N1 <> 0 Then
            AR(J1) = Round(AR(J1), 4)
            AR(J2) = Round(AR(J2), 4)
            List1.AddItem J & "     " & AR(J1) & "     " & AR(J2)
        End If
    Next J
    cmdResult.Enabled = False
    Exit Sub
Err_Hand:
    MsgBox "Please  input  data  again", vbOKOnly, "ERROR  DATA"
End Sub
```

12. โปรแกรมย่อย cmdStruc

```
Private Sub cmdStruc_Click()
        On Error GoTo Err_Hand
        imgScreen.Visible = False
        List1.Visible = True
        List1.Clear
        M = InputBox("Input number of member", "Structure Data", "M", 6174.1, 1420)
        NR = InputBox("Input number of support restraints", "Structure Data", "NR", _
              6174.1, 1420)
        NRJ = InputBox("Input number of restrainted joints", "Structure Data", "NRJ", _
               6174.1, 1420)
        E = InputBox("Input modulus of elasticity", "Structure Data", "E", 6174.1, 1420)
        NJ = M + 1
        ND = 2 * NJ
        N = ND - NR
        List1.AddItem "=================="
        List1.AddItem "Structural Data"
        List1.AddItem "=================="
        List1.AddItem "Number of members = " & M
        List1.AddItem "Number of support restraints = " & NR
        List1.AddItem "Number of restrainted joints = " & NRJ
        List1.AddItem "Modulus of elasticity = " & E
        List1.AddItem "Number of joints = " & NJ
        List1.AddItem "Number of degree of freedom = " & N
        cmdMemIn.Enabled = True
        Exit Sub
Err_Hand:
        MsgBox "Please input data again", vbOKOnly, "ERROR DATA"
End Sub
```

13. โปรแกรมย่อย cmdTriL

```
Private Sub cmdTriL_Click()
        Dim K As Integer, I As Integer
        On Error GoTo Err_Hand
        imgTri.Visible = True
        List1.AddItem "==================="
        List1.AddItem "Triangular  Load"
        List1.AddItem "==================="
        List1.AddItem "Member    TYPE    WT"
        For K = 1 To NTL
                I = InputBox("Input  member  index", "Triangular  Load", "MEMBER", _
                   6174.1, 1420)
                TYP = InputBox("Input  type  of  triangular  load", "Triangular  Load", "TYPE", _
                      6174.1,  1420)
                WT = InputBox("Input  WT", "Triangular  Load", "WT", 6174.1, 1420)
                List1.AddItem I & "      " & TYP & "      " & "      " & WT
                Select Case TYP
                        Case 1
                                ML(1, I) = AML(1, I) + (-WT * EL(I) / 4)
                                AML(2, I) = AML(2, I) + (-5 * WT * EL(I) ^ 2 / 96)
                                AML(3, I) = AML(3, I) + (-WT * EL(I) / 4)
                                AML(4, I) = AML(4, I) + (5 * WT * EL(I) ^ 2 / 96)
                        Case 2
                                AML(1, I) = AML(1, I) + (-3 / 20 * WT * EL(I))
                                AML(2, I) = AML(2, I) + (-WT * EL(I) ^ 2 / 30)
                                AML(3, I) = AML(3, I) + (-7 / 20 * WT * EL(I))
                                AML(4, I) = AML(4, I) + (WT * EL(I) ^ 2 / 20)
                        Case 3
                                AML(1, I) = AML(1, I) + (-7 / 20 * WT * EL(I))
                                AML(2, I) = AML(2, I) + (-WT * EL(I) ^ 2 / 20)
```

```
                    AML(3, I) = AML(3, I) + (-3 / 20 * WT * EL(I))
                    AML(4, I) = AML(4, I) + (WT * EL(I) ^ 2 / 30)
            End Select
            LML(I) = 1
        Next K
        cmdTriL.Enabled = False
        cmdResult.Enabled = True
        imgTri.Visible = False
        Exit Sub
    Err_Hand:
        MsgBox "Please input 'Load Data' again", vbOKOnly, "ERROR DATA"
        imgTri.Visible = False
        Image1.Visible = False
        cmdPoint.Visible = False
        cmdUnif.Visible = False
        cmdTriL.Visible = False
        cmdMMT.Visible = False
    End Sub
```

14. โปรแกรมย่อย cmdUnif

```
Private Sub cmdUnif_Click()
        Dim I As Integer, K As Integer
        On Error GoTo Err_Hand
        imgUnif.Visible = True
        List1.AddItem "=================="
        List1.AddItem "Uniform  Load"
        List1.AddItem "=================="
        List1.AddItem "Member     WU"
        For K = 1 To NUL
                I = InputBox("Input  member  index", "Uniform  Load", "MEMBER", _
                    6174.1, 1420)
                WU = InputBox("Input  WU", "Uniform  Load", "WU", 6174.1, 1420)
                List1.AddItem I & "      " & WU
                AML(1, I) = AML(1, I) + (-WU * EL(I) / 2)
                AML(2, I) = AML(2, I) + (-WU * EL(I) ^ 2 / 12)
                AML(3, I) = AML(3, I) + (-WU * EL(I) / 2)
                AML(4, I) = AML(4, I) + (WU * EL(I) ^ 2 / 12)
                LML(I) = 1
        Next K
        cmdUnif.Enabled = False
        cmdResult.Enabled = True
        imgUnif.Visible = False
        Exit Sub
Err_Hand:
        MsgBox "Please  input  'Load  Data'  again", vbOKOnly, "ERROR  DATA"
        imgUnif.Visible = False
        Image1.Visible = False
        cmdPoint.Visible = False
        cmdUnif.Visible = False
        cmdTriL.Visible = False
        cmdMMT.Visible = False
End Sub
```

15. โปรแกรมย่อย Form

```
Private Sub Form_Load()
        cmdMemIn.Enabled = False
        cmdJointRe.Enabled = False
        cmdLoadD.Enabled = False
        cmdResult.Enabled = False
        Image1.Visible = False
        cmdPoint.Visible = False
        cmdUnif.Visible = False
        cmdTriL.Visible = False
        cmdMMT.Visible = False
        imgDJ.Visible = False
        imgJL.Visible = False
        imgMMT.Visible = False
        imgPoint.Visible = False
        imgRes.Visible = False
        imgTri.Visible = False
        imgUnif.Visible = False
        List1.Visible = False
        List1.Clear
End Sub
```

16. โปรแกรมย่อย mnuNew

```
Private Sub mnuNew_Click()
        List1.Clear
End Sub
```

17. โปรแกรมย่อย mnuQuit

```
Private Sub mnuQuit_Click()
        frmBeam.Hide
        frmMain.Show
End Sub
```