

## ภาคผนวก

ในภาคผนวกนี้จะรวบรวม CODE ของทุกโปรแกรมย่อย โดยจะเรียงรายชื่อตามลำดับ  
อักษรของชื่อ โปรแกรม

### 1. การประกาศตัวแปรทั่วไปที่ใช้ในทุกโปรแกรมย่อย

Option Explicit

Option Base 1

Private M As Integer, NR As Integer, NRJ As Integer, E As Single, EL() As Single

Private X() As Single, Y() As Single, CX() As Single, CY() As Single

Private JJ() As Integer, JK() As Integer, XCL() As Integer, YCL() As Integer

Private AX() 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, AC()  
As Single

Private ID() As Integer, LML() As Integer, NJ As Integer

Private DJ() As Single, AM() As Single, AR() As Single

Private NBI As Integer, NB As Integer

Const MD = 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 = 1 - 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 I  
End Sub
```



#### 4. โปรแกรมย่อย cmdJointCo

```
Private Sub cmdJointCo_Click()
```

```
    Dim J As Integer, I As Integer
```

```
    On Error GoTo Err_Hand
```

```
    ReDim X(NJ)
```

```
    ReDim Y(NJ)
```

```
    List1.AddItem "—————"
```

```
    List1.AddItem "Joint coordinate"
```

```
    List1.AddItem "—————"
```

```
    List1.AddItem " J  X(J)  Y(J)"
```

```
    For J = 1 To NJ
```

```
        I = InputBox("Input joint index")
```

```
        X(I) = InputBox("Input X(I) coordinate", "Joint coordinate", "X" & I)
```

```
        Y(I) = InputBox("Input Y(I) coordinate", "Joint coordinate", "Y" & I)
```

```
        List1.AddItem I & "    " & X(I) & "    " & Y(I)
```

```
    Next J
```

```
    cmdMemIn.Enabled = True
```

```
    Exit Sub
```

```
Err_Hand:
```

```
    MsgBox "Please input 'Joint coordinate' again", vbOKOnly, "ERROR DATA"
```

```
End Sub
```

## 5. 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)

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")

JRL(2 \* K - 1) = InputBox("Input JR1", "Joint Restraints", "JR1")

JRL(2 \* K) = InputBox("Input JR2", "Joint Restraints", "JR2")

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

ID(J) = N + N1

Else

ID(J) = J - N1

End If

Next J

cmdLoadD.Enabled = True

Exit Sub

Err\_Hand:

MsgBox "Please input 'Joint Restraints' again", vbOKOnly, "ERROR DATA"

End Sub





## 6. โปรแกรมย่อย cmdLoadD

```

Private Sub cmdLoadD_Click()
    Dim J As Integer, K As Integer, JR As Integer
    On Error GoTo Err_Hand

    NLJ = InputBox("Input number of load joints", "Load Data", "NLJ")
    ReDim AC(ND)
    ReDim AJ(ND)
    For J = 1 To ND
        AJ(J) = 0
    Next J
    List1.AddItem "====="
    List1.AddItem "Action At Joints"
    List1.AddItem "====="
    List1.AddItem "Joint AJ1 AJ2"
    For J = 1 To NLJ
        K = InputBox("Input joint index", "Action At Joints", "JOINT")
        AJ(2 * K - 1) = InputBox("Input AJ1", "Action At Joint", "AJ1")
        AJ(2 * K) = InputBox("Input AJ2", "Action At Joint", "AJ2")
        List1.AddItem K & " " & AJ(2 * K - 1) & " " & AJ(2 * K)
    Next J
    For J = 1 To ND
        JR = ID(J)
        AC(JR) = AJ(J)
    Next J

    cmdResult.Enabled = True

    Exit Sub
Err_Hand:
    MsgBox "Please input 'Load Data' again", vbOKOnly, "ERROR DATA"
End Sub

```

## 7. โปรแกรมย่อย cmdMemIn

```
Private Sub cmdMemIn_Click()
```

```
    Dim J As Integer, I As Integer
```

```
    On Error GoTo Err_Hand
```

```
    NB = 0
```

```
    ReDim EL(M)
```

```
    ReDim AX(M)
```

```
    ReDim JJ(M)
```

```
    ReDim JK(M)
```

```
    ReDim XCL(M)
```

```
    ReDim YCL(M)
```

```
    ReDim CX(M)
```

```
    ReDim CY(M)
```

```
    List1.AddItem "_____"
```

```
    List1.AddItem "Member Information"
```

```
    List1.AddItem "_____"
```

```
    List1.AddItem "Member From joint To joint AX Length CX CY"
```

```
    For J = 1 To M
```

```
        I = InputBox("Input member index", "Member Information", "MEMBER")
```

```
        JJ(I) = InputBox("Input from joint", "Member Information", "JJ" & I)
```

```
        JK(I) = InputBox("Input to joint", "Member Information", "JK" & I)
```

```
        AX(I) = InputBox("Input cross-sectional area" & I, "Member Information", "AX")
```

```
        XCL(I) = X(JK(I)) - X(JJ(I))
```

```
        YCL(I) = Y(JK(I)) - Y(JJ(I))
```

```
        EL(I) = Sqr(XCL(I) ^ 2 + YCL(I) ^ 2)
```

```
        CX(I) = XCL(I) / EL(I)
```

```
        CY(I) = YCL(I) / EL(I)
```

```
        List1.AddItem I & "          " & JJ(I) & "          " & JK(I) & "          " & AX(I) & "
```

```
        " & EL(I) & " " & CX(I) & " " & CY(I)
```

```
        NBI = 2 * (Abs(JK(I) - JJ(I)) + 1)
```

```
If NBI > NB Then NB = NBI
```

```
Next J
```

```
cmdJointRe.Enabled = True
```

```
Exit Sub
```

```
Err_Hand:
```

```
MsgBox "Please input 'Member Information' again", vbOKOnly, "ERROR DATA"
```

```
End Sub
```



## 8. 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 Restrained 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 Cross section Area"
```

```
For I = 1 To M
```

```
Printer.Print I & " " & EL(I) & " " & AX(I)
```

```
Next I
```

```
Printer.Print "=====
```

```
Printer.Print "Restraint AT Joint"
```

```
Printer.Print "=====
```

```
Printer.Print "Joint 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 "Number Of Loaded Joints = " & NLJ
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
Printer.Print "===== "
Printer.Print "Result"
Printer.Print "===== "
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
```



### 9. โปรแกรมย่อย cmdQuit

```
Private Sub cmdQuit_Click()
```

```
    frmTruss.Hide
```

```
    frmMain.Show
```

```
End Sub
```



## 10. โปรแกรมย่อย cmdResult

Private Sub cmdResult\_Click()

Dim SCM() As Single, II As Integer, I2 As Integer, I As Integer, J As Integer, K As Integer

Dim IR As Integer, IC As Integer, SMS(4, 4) As Single, IM(4) As Integer, ITEM As Integer

Dim JR As Integer, DF() As Single, JE As Integer, AMD(4) As Single, SFF() As Single

Dim J1 As Integer, J2 As Integer, K1 As Integer, K2 As Integer, NI As Integer

Dim J10 As Integer, J11 As Integer, K11 As Integer

ReDim SCM(M)

ReDim SFF(N, NB)

ReDim AR(ND)

For J = 1 To ND

AR(J) = 0

Next J

For J10 = 1 To N

For K = 1 To NB

SFF(J10, K) = 0

Next K

Next J10

For I = 1 To M

SCM(I) = E \* AX(I) / EL(I)

SMS(1, 1) = SCM(I) \* CX(I) \* CX(I)

SMS(1, 2) = SCM(I) \* CX(I) \* CY(I)

SMS(I, 3) = -SMS(1, 1)

SMS(1, 4) = -SMS(1, 2)

SMS(2, 2) = SCM(I) \* CY(I) \* CY(I)

SMS(2, 3) = -SMS(1, 2)

SMS(2, 4) = -SMS(2, 2)

SMS(3, 3) = SMS(1, 1)

SMS(3, 4) = SMS(1, 2)

SMS(4, 4) = SMS(2, 2)



```

IM(1) = 2 * JJ(I) - 1
IM(2) = 2 * JJ(I)
IM(3) = 2 * JK(I) - 1
IM(4) = 2 * JK(I)
For J11 = 1 To MD
  II = IM(J11)
  If JRL(I1) <= 0 Then
    For K11 = J11 To MD
      I2 = IM(K11)
      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) + SMS(J11, K11)
      End If
    Next K11
  End If
Next J11
Next I
Call BANFAC(N, NB, SFF())
ReDim AC(ND)
For J = 1 To ND
  JR = ID(J)
  AC(JR) = AJ(J)
Next J

```

```

ReDim DF(N)
ReDim DJ(ND)
Call BANSOL(N, NB, SFF0, AC0, DF0)
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 "Result"
List1.AddItem "====="
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
    J1 = 2 * JJ(I) - 1

```

```

J2 = 2 * JJ(1)
K1 = 2 * JK(1) - 1
K2 = 2 * JK(1)
SCM(1) = E * AX(1) / EL(1)
AMD(1) = SCM(1) * ((DJ(J1) - DJ(K1)) * CX(1) + (DJ(J2) - DJ(K2)) * CY(1))
AMD(3) = -AMD(1)
AMD(2) = 0
AMD(4) = 0
For J = 1 To MD
    AM(J, 1) = AMD(J)
Next J
List1.AddItem I & " " & AM(1, 1) & " " & AM(2, 1) & " " & AM(3, 1) & " " & AM(4,
I)
    If JRL(J1) = 1 Then
        AR(J1) = AR(J1) + CX(1) * AMD(1)
    End If
    If JRL(J2) = 1 Then
        AR(J2) = AR(J2) + CY(1) * AMD(1)
    End If
    If JRL(K1) = 1 Then
        AR(K1) = AR(K1) + CX(1) * AMD(3)
    End If
    If JRL(K2) = 1 Then
        AR(K2) = AR(K2) + CY(1) * AMD(3)
    End If
Next I
For J = 1 To ND
    If JRL(J) <> 0 Then
        AR(J) = AR(J) - AJ(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
```

## 11. โปรแกรมย่อย cmdStruc

Private Sub cmdStruc\_Click()

On Error GoTo Err\_Hand

List1.Visible = True

List1.Clear

M = InputBox("Input number of member", "Structure Data", "M")

NJ = InputBox("Input number of joints", "Structure Data", "NJ")

NR = InputBox("Input number of support restraints", "Structure Data", "NR")

NRJ = InputBox("Input number of restrained joints", "Structure Data", "NRJ")

E = InputBox("Input modulus of elasticity", "Structure Data", "E")

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 restrained joints=" & NRJ

List1.AddItem "Modulus of elasticity=" & E

List1.AddItem "Number of joints=" & NJ

List1.AddItem "Number of degree of freedom=" & N

cmdJointCo.Enabled = True

Exit Sub

Err\_Hand:

MsgBox "Please input 'Structure data' again", vbOKOnly, "ERROR DATA"

End Sub