เอกสารอ้างอิง

1. นิคม ปราชญ์นคร, กฎหมายและระเบียบควบคุมอาคารพร้อมด้วยกฎหมายอื่น ๆ ที่เกี่ยวข้อง, วิศวกรรมสถานแห่งประเทศไทย ในพระบรมราชูปถัมภ์, พิมพ์ครั้งที่ 2, เมษายน 2528.
2. Davenport, A.G., "Rational for Determining Design Wind Velocities," Journal of The Structural Division Proceedings, ASCE., Vol.85, ST.5, Proc. Paper 3123, May 1960.
3. Davenport, A.G., Hartmann, A.J., "Comparison of The Predicted and Measured Dynamic Response of Structures to Wind (CFPL. T.V. Tower)," Engineering Science, The University of Western Ontario, London, Canada, 1966.
4. Estoque, A.F., "Spectral Analysis of Simulates Wind Records Including Time-Varying Mean and Gusts," M. Eng. Thesis, AIT. Bangkok, 1968.
5. Simiu, E., "Logarithmic Profiles and Design Wind Speeds," Journal of The Engineering Mechanics Division, ASCE., pp. 1074-1083, October 1973.
6. Pannachet, Y., "Simulated Wind for Structural Dynamic Analysis in Bangkok," M. Eng. Thesis, AIT., Bangkok, 1975.
7. Bathe, K.J., Wilson, E.L., "Numerical Methods in Finite Element Analysis," Prentice-Hall, Inc., Englewood Cliffs, N.J., 1976.
8. Bathe, K.J.,"Solution Method of Large Generalized Eigenvalue Problem in Structural Engineering," Report UC SESM 71-20, Civil Engineering Department, University of California, Berkeley, 1971.
9. Wilson, E.L., Bathe, K.J. and Doherty, W.P.,"Direct Solution of Large System of Linear Equations," Computers and Structures, Vol.4 No.2, 1974, pp. 363-372.
10. Clough, R.W., Josep Penzien, "Dynamics of Structures," McGraw-Hill, 1975.

11. อุตุนิยมวิทยา, กรม, ข้อมูลกราฟลมมุนโรไดน์, ถนนสุขุมวิท, กรุงเทพมหานคร, พ.ศ. 2512-2519.

12. Gaylord, E.H., Jr., Gaylord, C.N., "Structural Engineering Handbook," McGraw-Hill, 1979.

13. American National Standard A58.1-1982, Building Code Requirements for Minimum Design Loads for Buildings and Other Structures, 1982.

14. Mario Paz, "Structural Dynamics Theory and Computation," Van Nostrand Reinhold, 1980.

15. Bathe, K.J., Wilson, E.L.and Perterson, F.E., "SAP IV - A Structural Analysis Program for Static and Dynanic Response of Linear Systems," Report EERC 73-11, College of Engineering, University of California, Berkeley, June 1973, revised Apr. 1974.

16. Task Committee on Wind Forces, "Wind Forces on Structures", Transaction ASCE., Vol.126, Part II, Paper No. 3269, pp. 1124-1198, 1961.

17. ทักษิณ เทพชาตรี, โปรแกรมไมโครคอมพิวเตอร์สำหรับวิเคราะห์โครงข้อแข็ง 2 มิติ, เลขที่ คส 04/2526 ภาควิชาวิศวกรรมโยธา จุฬาลงกรณ์มหาวิทยาลัย.

ภาคผนวก

ภาคผนวก ก.

วิธีการป้อนข้อมูลและการแสดงตัวอย่างข้อมูล

ก-1 ความนำ

โปรแกรมที่ใช้สำหรับวิเคราะห์โครงสร้างชนิดโครงของแข็ง 2 มิติ แบบพลศาสตร์นี้ เขียนขึ้นเพื่อใช้กับเครื่องไมโครคอมพิวเตอร์ IBM โดยเขียนโปรแกรมเป็นภาษาเบสิค การนำโปรแกรมนี้ไปใช้กับเครื่องไมโครคอมพิวเตอร์ชนิดอื่น คำสั่งบางคำสั่งที่ใช้ในโปรแกรมอาจจะต้องเปลี่ยนแปลงใหม่ให้เหมาะสมกับเครื่องชนิดนั้น ๆ รายละเอียดของโปรแกรมสามารถดูได้จากภาคผนวก ข.

ก-2 ลำดับของการป้อนข้อมูล

การป้อนข้อมูลจะกระทำตามลำดับต่าง ๆ ดังต่อไปนี้

1. ป้อนข้อมูลเกี่ยวกับชื่อข้อมูล

2. ป้อนข้อมูลเกี่ยวกับคุณสมบัติทั่วไปของโครงข้อแข็ง

2.1 โมดูลัสของการยืดหยุ่นเท่ากันทั้งโครงสร้างหรือไม่ โดยตอบ Y(es) หรือ N(o)

2.2 จำนวนชิ้นส่วนทั้งหมด (NM)

2.3 จำนวนข้อทั้งหมด (NJ)

2.4 จำนวนรูปแบบของแรงกระทำ (NH) ไม่เกิน 4 รูปแบบ

2.5 จำนวนชิ้นส่วนที่มีแรงกระทำ (ML)

2.6 จำนวนข้อต่อที่มีแรงกระทำ (JL)

2.7 จำนวนฐานรองรับ (JBC)

2.8 ค่าของโมดูลัสของการยืดหยุ่น (E) เมื่อข้อ 2.1 ตอบ Y(es)

2.9 ตอบคำถามว่ายอมรับข้อมูลชุดนี้หรือไม่ โดยตอบ Y(es) หรือ N(o)

3. ป้อนข้อมูลเกี่ยวกับตำแหน่งของชั้ว

3.1 ตำแหน่งของชั้วในแนวราบ (X(I))

3.2 ตำแหน่งของชั้วในแนวดิ่ง (Y(I))

3.3 ตอบคำถามที่ยอมรับข้อมูลชุดนี้หรือไม่ โดยตอบ Y(es) หรือ N(o)

4. ป้อนข้อมูลการเชื่อมต่อและคุณสมบัติของชิ้นส่วน

4.1 หมายเลขชั้วทางด้านซ้ายของชิ้นส่วน (J(I))

4.2 หมายเลขชั้วทางด้านขวาของชิ้นส่วน (K(I))

4.3 พื้นที่หน้าตัดของชิ้นส่วน (AR(I))

4.4 โมเมนต์อินเนอร์เชียของชิ้นส่วน (AINER(I))

4.5 มวลต่อความยาวของชิ้นส่วน (AM(I)) เมื่อรูปแบบของแรงกระทำมากกว่าศูนย์

4.6 โมดูลัสของการยืดหยุ่นของชิ้นส่วน (E(I)) เมื่อข้อ 2.1 ตอบ N(o)

4.7 ตอบคำถามที่ยอมรับข้อมูลชุดนี้หรือไม่ โดยตอบ Y(es) หรือ N(o)

5. ป้อนข้อมูลของหมายเลขข้อต่อที่เป็นฐานรองรับ (JB(I))

6. ป้อนข้อมูลเกี่ยวกับรูปแบบของแรงกระทำ

6.1 ตอบคำถามว่าแรงลมหรือไม่ โดยตอบ Y(es) หรือ N(o)

6.2 จำนวนจุดของรูปแบบของแรงกระทำ (NT(J))

6.3 เวลาที่จุดต่าง ๆ (T(J,I))

6.4 ค่าแรงกระทำที่จุดนั้น ๆ (V(J,I))

6.5 ตอบคำถามที่ยอมรับข้อมูลชุดนี้หรือไม่ โดยตอบ Y(es) หรือ N(o)

7. ป้อนข้อมูลเกี่ยวกับแรงกระทำบนชิ้นส่วน (ถ้ามี)

7.1 หมายเลขชิ้นส่วนที่มีแรงกระทำ (K1(I))

7.2 ค่าแรงกระจายในแนวดิ่ง หรือตัวประกอบของแรงกระทำ (W(I,1))

7.3 หมายเลขรูปแบบของแรงกระทำ (NW(I,1)) เมื่อรูปแบบของแรงกระทำมากกว่าศูนย์

7.4 ค่าแรงกระจายในแนวราบ หรือตัวประกอบของแรงกระทำ (W(I,2))

7.5 หมายเลขรูปแบบของแรงกระทำ (NW(I,2)) เมื่อรูปแบบของแรงกระทำมากกว่าศูนย์

7.6 จำนวนแรงแบบจุดในแนวดิ่ง (P(I,2))

7.7 จำนวนแรงแบบจุดในแนวราบ (P(I,1))

7.8 ป้อนข้อมูลเกี่ยวกับแรงกระทำแบบจุดในแนวดิ่ง (ถ้ามี)

7.8.1 ค่าแรงกระทำ หรือตัวประกอบของแรงกระทำ (V1(J,I))

7.8.2 ระยะที่แรงกระทำจากปลายด้านซ้ายมือ (VA(J,I))

7.8.3 หมายเลขรูปแบบของแรงกระทำ (NV(J,1)) เมื่อรูปแบบของแรงกระทำมากกว่าศูนย์

7.9 ป้อนข้อมูลเกี่ยวกับแรงกระทำแบบจุดในแนวราบ (ถ้ามี)

7.9.1 ค่าแรงกระทำ หรือตัวประกอบของแรงกระทำ (H1(J,I))

7.9.2 ระยะที่แรงกระทำจากปลายด้านซ้ายมือ (HA(J,I))

7.9.3 หมายเลขรูปแบบของแรงกระทำ (NH(J,I)) เมื่อรูปแบบของแรงกระทำมากกว่าศูนย์

7.10 ตอบคำถามที่ยอมรับข้อมูลชุดนี้หรือไม่ โดยตอบ Y(es) หรือ N(o)

8. ป้อนข้อมูลเกี่ยวกับแรงกระทำที่ข้อต่อ (ถ้ามี)

8.1 หมายเลขข้อต่อที่แรงกระทำ (K2(I))

8.2 ค่าแรงกระทำในแนวราบ (PJ(I,1))

8.3 หมายเลขรูปแบบของแรงกระทำ (NP(I,1)) เมื่อรูปแบบของแรงกระทำมากกว่าศูนย์

8.4 ค่าแรงกระทำในแนวดิ่ง (PJ(I,2))

8.5 หมายเลขรูปแบบของแรงกระทำ (NP(I,2)) เมื่อรูปแบบของแรงกระทำมากกว่าศูนย์

8.6 ค่าโมเมนต์ (PJ(I,3))

8.7 หมายเลขรูปแบบของแรงกระทำ (NP(I,3)) เมื่อรูปแบบของแรงกระทำมากกว่าศูนย์

8.8 ตอบคำถามที่ยอมรับข้อมูลชุดนี้หรือไม่ โดยตอบ Y(es) หรือ N(o)

9. ป้อนข้อมูลอื่น ๆ เมื่อเป็นการคำนวณโดยทฤษฎีพลศาสตร์

9.1 จำนวนรูปแบบโหมดที่สำคัญ (NNC)

9.2 ค่าของอัตราการสูญเสียพลังงาน (XI)

9.3 พารามิเตอร์ของกฎแห่งกำลัง (PLP)

9.4 ช่วงเวลาที่ทำการอินทิเกรต (DT)

9.5 เวลามากที่สุดที่ทำการอินทีเกรต (TMAX)

9.6 ตอบคำถามที่ยอมรับข้อมูลชุดนี้หรือไม่ โดยตอบ Y(es) หรือ N(o)

ก-4 <u>ตัวอย่างป้อนข้อมูล</u>

แสดงตัวอย่างการป้อนข้อมูลในตัวอย่างที่ 1 (หัวข้อ 5.2)

```
**********************************************
**  STATIC & DYNAMIC ANALYSIS OF 2-D FRAME  **
**********************************************

        1> INPUT DATA

        2> DISPLAY & EDIT DATA

        3> ANALYZE

        4> EXIT

 ENTER SELECTION NO.<1-4> ? 1


 INPUT DATA DISK INTO DRIVE A:

 NAME OF DATA FILE TO BE SAVED

 FILE NAME OF STRUCTURE.......? EXAMPLE1
```

```
**********************
**   INPUT DATA   **
**********************

IS YOUNG'S MODULUS CONSTANT ? (Y/N) ? Y

NO. OF MEMBER.............= ? 2

NO. OF NODES..  ..........= ? 3

NO. OF HISTORY LOAD........
 [EQ.0 => STATIC ANA.].....
 [GE.1 => DYANMIC ANA.]...= ? 1

NO. OF MEMBERS LOAD.......= ? 0

NO. OF JOINTS LOAD........= ? 1

NO. OF BOUNDARY NODES.....= ? 2

YOUNG'S MODULUS...........= ? 10000000

DO YOU ACCEPT THESE DATA ? (Y/N) ? Y


-----------------------------------------
    INPUT NODAL POINT COORDINATES
-----------------------------------------

 NODE  NO.  1
 =============

 X-COORDINATE.........= ? 0
 Y-COORDINATE.........= ? 0

DO YOU ACCEPT THESE DATA ? (Y/N) ? Y


-----------------------------------------
    INPUT NODAL POINT COORDINATES
-----------------------------------------

 NODE  NO.  2
 =============

 X-COORDINATE.........= ? 70.710678
 Y-COORDINATE.........= ? 70.710678

DO YOU ACCEPT THESE DATA ? (Y/N) ? Y
```

```
-----------------------------------------
    INPUT NODAL POINT COORDINATES
-----------------------------------------

 NODE  NO.  3
 =============

 X-COORDINATE.........= ? 170.710678
 Y-COORDINATE.........= ? 70.710678

DO YOU ACCEPT THESE DATA ? (Y/N) ? Y


-----------------------------------------
    INPUT NODE NO. & PROP. OF MEMBERS
-----------------------------------------

 MEMBER NO.  1
 =============

 NODE J..............= ? 1
 NODE K..............= ? 2
 CROSS SECTIONL AREA..= ? 6
 MOMENT OF INERTIA....= ? 100
 MASS PER UNIT LENGTH.= ? 4.2

DO YOU ACCEPT THESE DATA ? (Y/N) ? Y


-----------------------------------------
    INPUT NODE NO. & PROP. OF MEMBERS
-----------------------------------------

 MEMBER NO.  2
 =============

 NODE J..............= ? 2
 NODE K..............= ? 3
 CROSS SECTIONL AREA..= ? 6
 MOMENT OF INERTIA....= ? 200
 MASS PER UNIT LENGTH.= ? 4.2

DO YOU ACCEPT THESE DATA ? (Y/N) ? Y


-----------------------------------------
INPUT BC. AT NODES ARE FIXED
-----------------------------------------

 NODE NO. ? 1
 NODE NO. ? 3

DO YOU ACCEPT THESE DATA ? (Y/N) ? Y
```

```
 ********************************************
 **  INPUT DATA OF HISTORY LOAD TYPE NO. 1  **
 ********************************************

IS THIS WIND HISTORY ? (Y/N) ? N
NO. OF POINT OF THIS FUNCTION ? 2

TIME VALUES AT POINT  1 ? 0
LOAD VALUES AT POINT  1 ? 100000

TIME VALUES AT POINT  2 ? .025
LOAD VALUES AT POINT  2 ? 100000

DO YOU ACCEPT THESE DATA ? (Y/N) ? Y



-----------------------------------------
 INPUT NODAL LOADS IN GLOBAL COORDINATES
-----------------------------------------

 NODE NO. ? 2
==================
 FOREC IN X-DIRECTION.............= ? 1
 HISTORY LODA TYPE NO.............= ? 1
 FOREC IN Y-DIRECTION.............= ? 0
 HISTORY LODA TYPE NO.............= ? 0
 MOMENT...........................= ? 0
 HISTORY LODA TYPE NO.............= ? 0

 DO YOU ACCEPT THESE DATA ? (Y/N) ? Y


-----------------------------------------
    INPUT OTHER DATA
-----------------------------------------

  NO. OF FREQUENCIES TO BE FOUND
      IN THE EIGENVALUE SOLUTION ? 3

  DAMPING FACTOR TO BE APPLIDE
                 TO ALL MODE.. ? 0
RUN
  POWER - LAW PARAMETER OF WIND. ? 0

  TIME STEP OF INTEGRATION..... ? .001

  MAX. TIME OF INTEGRATION..... ? .02


 DO YOU ACCEPT THESE DATA ? (Y/N) ? Y
```

ก-5 ตัวอย่างผลลัพธ์

แสดงตัวอย่างผลลัพธ์ในตัวอย่างที่ 1 (หัวข้อ 5.2)

```
**********************************************
**  STATIC & DYNAMIC ANALYSIS OF 2-D FRAME  **
**********************************************

          1> INPUT DATA

          2> DISPLAY & EDIT DATA

          3> ANALYZE

          4> EXIT

     ENTER SELECTION NO.<1-4> ? 3

FILE NAME OF STRUCTURE.......? EXAMPLE1

 RUNNING STIFFNESS & MASS

 RUNNING STRUCTURE

 RUNING DECOMPOSE

SUBSPACE ITERRATION METHOD

CONVERGENCE REACHED FOR RTOL =  .00001

 THE CALCULATED EIGENVALUES ARE
 ==============================
EIGV( 1 ) =  638.5045874481654
EIGV( 2 ) =  976.5825465395022
EIGV( 3 ) =  4211.597108802474
```

THE CALCULATED EIGENVECTORS ARE
==========

R(1 , 1) = -2.1829746318966960-02
R(2 , 1) = 5.270166975820494D-02
R(3 , 1) = 3.534860583269087D-12
R(1 , 2) = 4.9788929483475610-03
R(2 , 2) = 2.062324859332281D-03
R(3 , 2) = 3.4092864445885240-03
R(1 , 3) = -5.830737142119192D-02
R(2 , 3) = -2.415170400340183D-02
R(3 , 3) = 1.629149869821614D-03

MODAL SUPERPOSITION MATHOD

PRINT OUT DATA

TIME = .001
==========

****
*** NODAL RESPONSE IN GLOBAL COORDINATES ***
****

NODE	DIRCETn.	DISPLACEMENT	VELOCITY	ACCERELATION
1	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000
2	X	0.0002	0.3899	389.3792
	Y	0.0000	0.0267	26.5399
	ROTATn.	-0.0000	-0.0078	-7.7824
3	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000

****
*** MEMBER FORCES IN LOCAL COORDINATES ***
****

MEMBER	NODE	AXIAL FORCE	SHEAR FORCE	MOMENT
1	1	-88.4054	-0.7985	-0.9328
	2	88.4054	0.7985	-78.9169
2	2	116.9955	-2.1790	-147.9396
	3	-116.9955	2.1790	-69.9555

TIME = .002

****
*** NODAL RESPONSE IN GLOBAL COORDINATES ***
****

NODE	DIRCETn.	DISPLACEMENT	VELOCITY	ACCERELATION
1	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000
2	X	0.0008	0.7783	387.1885
	Y	0.0001	0.0529	25.7479
	ROTATn.	-0.0000	-0.0156	-7.7255
3	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000

****
*** MEMBER FORCES IN LOCAL COORDINATES ***
****

MEMBER	NODE	AXIAL FORCE	SHEAR FORCE	MOMENT
1	1	-353.2002	-3.1864	-3.5450
	2	353.2002	3.1864	-315.0990
2	2	467.5454	-8.7075	-591.1529
	3	-467.5454	8.7075	-279.5989

TIME = .003

****
*** NODAL RESPONSE IN GLOBAL COORDINATES ***
****

NODE	DIRCETn.	DISPLACEMENT	VELOCITY	ACCERELATION
1	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000
2	X	0.0018	1.1637	383.5176
	Y	0.0001	0.0781	24.4909
	ROTATn.	-0.0000	-0.0232	-7.6292
3	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000

****
*** MEMBER FORCES IN LOCAL COORDINATES ***
****

MEMBER	NODE	AXIAL FORCE	SHEAR FORCE	MOMENT
1	1	-793.1201	-7.1415	-7.2914
	2	793.1201	7.1415	-706.8603
2	2	1050.3348	-19.5609	-1327.8315
	3	-1050.3348	19.5609	-628.2625

TIME = .004

****
*** NODAL RESPONSE IN GLOBAL COORDINATES ***
****

NODE	DIRCETn.	DISPLACEMENT	VELOCITY	ACCERELATION
1	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000
2	X	0.0031	1.5449	378.4652
	Y	0.0002	0.1017	22.6648
	ROTATn.	-0.0001	-0.0308	-7.4971
3	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000

****
*** MEMBER FORCES IN LOCAL COORDINATES ***
****

MEMBER	NODE	AXIAL FORCE	SHEAR FORCE	MOMENT
1	1	-1406.0744	-12.6264	-11.2680
	2	1406.0744	12.6264	-1251.3749
2	2	1863.1651	-34.6967	-2354.8893
	3	-1863.1651	34.6967	-1114.7824

TIME = .005

****
*** NODAL RESPONSE IN GLOBAL COORDINATES ***
****

NODE	DIRCETn.	DISPLACEMENT	VELOCITY	ACCERELATION
1	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000
2	X	0.0048	1.9202	371.9523
	Y	0.0003	0.1232	20.3525
	ROTATn.	-0.0001	-0.0382	-7.3264
3	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000

****
*** MEMBER FORCES IN LOCAL COORDINATES ***
****

MEMBER	NODE	AXIAL FORCE	SHEAR FORCE	MOMENT
1	1	-2189.1452	-19.5893	-14.2028
	2	2189.1452	19.5893	-1944.7252
2	2	2903.0020	-54.0573	-3668.1280
	3	-2903.0020	54.0573	-1737.6056

TIME = .006

****
*** NODAL RESPONSE IN GLOBAL COORDINATES ***
****

NODE	DIRCETn.	DISPLACEMENT	VELOCITY	ACCERELATION
1	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000
2	X	0.0069	2.2883	364.0834
	Y	0.0005	0.1422	17.5426
	ROTATn.	-0.0001	-0.0454	-7.1208
3	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000

****
*** MEMBER FORCES IN LOCAL COORDINATES ***
****

MEMBER	NODE	AXIAL FORCE	SHEAR FORCE	MOMENT
1	1	-3138.5911	-27.9634	-14.4657
	2	3138.5911	27.9634	-2781.8720
2	2	4165.9420	-77.5683	-5262.1166
	3	-4165.9420	77.5683	-2494.7103

TIME = .007

****
*** NODAL RESPONSE IN GLOBAL COORDINATES ***
****

NODE	DIRCETn.	DISPLACEMENT	VELOCITY	ACCERELATION
1	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000
2	X	0.0094	2.6479	354.8023
	Y	0.0006	0.1582	14.3158
	ROTATn.	-0.0002	-0.0524	-6.8784
3	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000

****
*** MEMBER FORCES IN LOCAL COORDINATES ***
****

MEMBER	NODE	AXIAL FORCE	SHEAR FORCE	MOMENT
1	1	-4249.8826	-37.6683	-10.0929
	2	4249.8826	37.6683	-3756.7381
2	2	5647.2622	-105.1398	-7130.3110
	3	-5647.2622	105.1398	-3383.6658

TIME = .008

****
*** NODAL RESPONSE IN GLOBAL COORDINATES ***
****

NODE	DIRCETn.	DISPLACEMENT	VELOCITY	ACCERELATION
1	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000
2	X	0.0122	2.9975	344.2828
	Y	0.0008	0.1707	10.5414
	ROTATn.	-0.0002	-0.0592	-6.6030
3	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000

****
*** MEMBER FORCES IN LOCAL COORDINATES ***
****

MEMBER	NODE	AXIAL FORCE	SHEAR FORCE	MOMENT
1	1	-5517.7144	-48.6100	1.1982
	2	5517.7144	48.6100	-4862.1979
2	2	7341.4010	-136.6654	-9264.9866
	3	-7341.4010	136.6654	-4401.5705

TIME = .009

****
*** NODAL RESPONSE IN GLOBAL COORDINATES ***
****

NODE	DIRCETn.	DISPLACEMENT	VELOCITY	ACCERELATION
1	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000
2	X	0.0154	3.3360	332.4037
	Y	0.0009	0.1792	6.3876
	ROTATn.	-0.0003	-0.0656	-6.2933
3	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000

****
*** MEMBER FORCES IN LOCAL COORDINATES ***
****

MEMBER	NODE	AXIAL FORCE	SHEAR FORCE	MOMENT
1	1	-6936.0242	-60.6806	22.0462
	2	6936.0242	60.6806	-6090.1051
2	2	9242.0424	-172.0250	-11657.3255
	3	-9242.0424	172.0250	-5545.1742

TIME = .01

****
*** NODAL RESPONSE IN GLOBAL COORDINATES ***
****

NODE	DIRCETn.	DISPLACEMENT	VELOCITY	ACCERELATION
1	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000
2	X	0.0189	3.6619	319.3646
	Y	0.0011	0.1833	1.7998
	ROTATn.	-0.0004	-0.0718	-5.9532
3	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000

****
*** MEMBER FORCES IN LOCAL COORDINATES ***
****

MEMBER	NODE	AXIAL FORCE	SHEAR FORCE	MOMENT
1	1	-8498.0141	-73.7598	55.3701
	2	8498.0141	73.7598	-7431.3511
2	2	11342.0673	-211.0828	-14297.5030
	3	-11342.0673	211.0828	-6810.7817

TIME = .011

****
*** NODAL RESPONSE IN GLOBAL COORDINATES ***
****

NODE	DIRCETn.	DISPLACEMENT	VELOCITY	ACCERELATION
1	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000
2	X	0.0227	3.9743	305.1462
	Y	0.0013	0.1826	-3.2721
	ROTATn.	-0.0004	-0.0775	-5.5816
3	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000

****
*** MEMBER FORCES IN LOCAL COORDINATES ***
****

MEMBER	NODE	AXIAL FORCE	SHEAR FORCE	MOMENT
1	1	-10196.2045	-87.7153	104.3875
	2	10196.2045	87.7153	-8875.9162
2	2	13633.6521	-253.6899	-17174.6489
	3	-13633.6521	253.6899	-8194.3452

TIME = .012

****
*** NODAL RESPONSE IN GLOBAL COORDINATES ***
****

NODE	DIRCETn.	DISPLACEMENT	VELOCITY	ACCERELATION
1	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000
2	X	0.0268	4.2718	289.7475
	Y	0.0015	0.1767	-8.5628
	ROTATn.	-0.0005	-0.0829	-5.1816
3	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000

****
*** MEMBER FORCES IN LOCAL COORDINATES ***
****

MEMBER	NODE	AXIAL FORCE	SHEAR FORCE	MOMENT
1	1	-12022.4190	-102.4019	172.6397
	2	12022.4190	102.4019	-10412.8304
2	2	16108.2664	-299.6841	-20276.9386
	3	-16108.2664	299.6841	-9691.4685

TIME = .013

****
*** NODAL RESPONSE IN GLOBAL COORDINATES ***
****

NODE	DIRCETn.	DISPLACEMENT	VELOCITY	ACCERELATION
1	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000
2	X	0.0313	4.5535	273.3831
	Y	0.0017	0.1653	-14.3470
	ROTATn.	-0.0006	-0.0879	-4.7553
3	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000

****
*** MEMBER FORCES IN LOCAL COORDINATES ***
****

MEMBER	NODE	AXIAL FORCE	SHEAR FORCE	MOMENT
1	1	-13967.8789	-117.6653	263.8482
	2	13967.8789	117.6653	-12030.3758
2	2	18756.6800	-348.8890	-23591.5642
	3	-18756.6800	348.8890	-11297.3402

TIME = .014

****
*** NODAL RESPONSE IN GLOBAL COORDINATES ***
****

NODE	DIRCETn.	DISPLACEMENT	VELOCITY	ACCERELATION
1	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000
2	X	0.0359	4.8183	256.0110
	Y	0.0018	0.1479	-20.4381
	ROTATn.	-0.0007	-0.0924	-4.3034
3	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000

****
*** MEMBER FORCES IN LOCAL COORDINATES ***
****

MEMBER	NODE	AXIAL FORCE	SHEAR FORCE	MOMENT
1	1	-16023.1921	-133.3396	382.0417
	2	16023.1921	133.3396	-13716.0004
2	2	21569.0733	-401.1184	-27104.9419
	3	-21569.0733	401.1184	-13006.8998

TIME = .015

****
*** NODAL RESPONSE IN GLOBAL COORDINATES ***
****

NODE	DIRCETn.	DISPLACEMENT	VELOCITY	ACCERELATION
1	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000
2	X	0.0409	5.0652	237.6692
	Y	0.0020	0.1243	-26.7271
	ROTATn.	-0.0008	-0.0965	-3.8281
3	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000

****
*** MEMBER FORCES IN LOCAL COORDINATES ***
****

MEMBER	NODE	AXIAL FORCE	SHEAR FORCE	MOMENT
1	1	-18178.4089	-149.2496	531.4672
	2	18178.4089	149.2496	-15456.4224
2	2	24535.0254	-456.1741	-30802.6521
	3	-24535.0254	456.1741	-14814.7625

TIME = .016

****
*** NODAL RESPONSE IN GLOBAL COORDINATES ***
****

NODE	DIRCETn.	DISPLACEMENT	VELOCITY	ACCERELATION
1	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000
2	X	0.0461	5.2934	218.6008
	Y	0.0021	0.0943	-33.3834
	ROTATn.	-0.0009	-0.1001	-3.3330
3	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000

****
*** MEMBER FORCES IN LOCAL COORDINATES ***
****

MEMBER	NODE	AXIAL FORCE	SHEAR FORCE	MOMENT
1	1	-20423.0820	-165.2122	716.5236
	2	20423.0820	165.2122	-17237.7420
2	2	27643.5415	-513.8468	-34669.4730
	3	-27643.5415	513.8468	-16715.2074

TIME = .017

****
*** NODAL RESPONSE IN GLOBAL COORDINATES ***
****

NODE	DIRCETn.	DISPLACEMENT	VELOCITY	ACCERELATION
1	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000
2	X	0.0515	5.5021	198.6653
	Y	0.0021	0.0575	-40.2713
	ROTATn.	-0.0010	-0.1031	-2.8170
3	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000

****
*** MEMBER FORCES IN LOCAL COORDINATES ***
****

MEMBER	NODE	AXIAL FORCE	SHEAR FORCE	MOMENT
1	1	-22746.2983	-181.0359	941.8399
	2	22746.2983	181.0359	-19045.4282
2	2	30883.1760	-573.9191	-38689.5885
	3	-30883.1760	573.9191	-18702.3204

TIME = .018

****
*** NODAL RESPONSE IN GLOBAL COORDINATES ***
****

NODE	DIRCETn.	DISPLACEMENT	VELOCITY	ACCERELATION
1	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000
2	X	0.0571	5.6905	178.1481
	Y	0.0022	0.0137	-47.2781
	ROTATn.	-0.0011	-0.1057	-2.2869
3	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000

****
*** MEMBER FORCES IN LOCAL COORDINATES ***
****

MEMBER	NODE	AXIAL FORCE	SHEAR FORCE	MOMENT
1	1	-25136.6895	-196.5220	1212.1958
	2	25136.6895	196.5220	-20864.3908
2	2	34241.9796	-636.1642	-42846.5055
	3	-34241.9796	636.1642	-20769.9189

TIME = .019

****
*** NODAL RESPONSE IN GLOBAL COORDINATES ***
****

NODE	DIRCETn.	DISPLACEMENT	VELOCITY	ACCERELATION
1	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000
2	X	0.0628	5.8581	156.9854
	Y	0.0022	-0.0371	-54.5067
	ROTATn.	-0.0012	-0.1077	-1.7412
3	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000

****
*** MEMBER FORCES IN LOCAL COORDINATES ***
****

MEMBER	NODE	AXIAL FORCE	SHEAR FORCE	MOMENT
1	1	-27582.5742	-211.4669	1532.4640
	2	27582.5742	211.4669	-22679.1540
2	2	37707.6403	-700.3483	-47123.2258
	3	-37707.6403	700.3483	-22911.6078

TIME = .02

****
*** NODAL RESPONSE IN GLOBAL COORDINATES ***
****

NODE	DIRCETn.	DISPLACEMENT	VELOCITY	ACCERELATION
1	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000
2	X	0.0688	6.0043	135.3101
	Y	0.0021	-0.0953	-61.9049
	ROTATn.	-0.0013	-0.1092	-1.1827
3	X	0.0000	0.0000	0.0000
	Y	0.0000	0.0000	0.0000
	ROTATn.	0.0000	0.0000	0.0000

****
*** MEMBER FORCES IN LOCAL COORDINATES ***
****

MEMBER	NODE	AXIAL FORCE	SHEAR FORCE	MOMENT
1	1	-30071.9131	-225.6616	1907.6455
	2	30071.9131	225.6616	-24473.8071
2	2	41267.4668	-766.2308	-51502.2644
	3	-41267.4668	766.2308	-25120.8119

****
** STATIC & DYNAMIC ANALYSIS OF 2-D FRAME **
****

1> INPUT DATA

2> DISPLAY & EDIT DATA

3> ANALYZE

4> EXIT

ENTER SELECTION NO.<1-4> ? 4

ภาคผนวก ข

รายละเอียดของโปรแกรมคอมพิวเตอร์

```
1000 DEFDBL A-H,O-Z : DEFINT I-N
1010 PRINT CHR$(12) : ' **  OPTION  **
1020 PRINT "  ***********************************************"
1030 PRINT "  **  STATIC & DYNAMIC ANALYSIS OF 2-D FRAME  **"
1040 PRINT "  ***********************************************"
1050 PRINT : PRINT : PRINT "             1> INPUT DATA"
1060 PRINT : PRINT "            2> DISPLAY & EDIT DATA"
1070 PRINT : PRINT "            3> ANALYZE"
1080 PRINT : PRINT "            4> EXIT"
1090 PRINT : INPUT "      ENTER SELECTION NO.<1-4> ";N1
1100 IF N1 < 1 OR N1 > 4 THEN 1090
1110 PRINT CHR$(12) : ON N1 GOTO 1120,1130,1140,1150
1120 CHAIN "INPUT.BAS"
1130 CHAIN "DISPLAY.BAS"
1140 CHAIN "STIFF.BAS"
1150 END
```

```
1000 DEFDBL A-Z :DEFINT I-N : PRINT CHR$(12)
1010 PRINT : PRINT "     INPUT DATA DISK INTO DRIVE A:" : PRINT
1020 PRINT "     NAME OF DATA FILE TO BE SAVED ":PRINT
1030 INPUT "     FILE NAME OF STRUCTURE.......";C$
1040 OPEN "O",#1,C$
1050 PRINT CHR$(12) : PRINT"  ********************"
1060 PRINT"  **   INPUT DATA   **"
1070 PRINT"  ********************"
1080 PRINT : INPUT "  IS YOUNG'S MODULUS CONSTANT ? (Y/N) ";F$
1090 PRINT : INPUT "  NO. OF MEMBER.............= ";NM
1100 PRINT : INPUT "  NO. OF NODES..  ..........= ";NJ
1110 PRINT : PRINT "  NO. OF HISTORY LOAD........ "
1120 PRINT "   [EQ.0 => STATIC ANA.]..... "
1130 INPUT "   [GE.1 => DYANMIC ANA.]...= ";NH
1140 PRINT : INPUT "  NO. OF MEMBERS LOAD.......= ";ML
1150 PRINT : INPUT "  NO. OF JOINTS LOAD........= ";JL
1160 PRINT : INPUT "  NO. OF BOUNDARY NODES.....= ";JBC
1170 IF F$ = "N" GOTO 1190
1180 PRINT : INPUT "  YOUNG'S MODULUS...........= ";E
1190 PRINT : INPUT " DO YOU ACCEPT THESE DATA ? (Y/N) ";B$
1200 IF B$ = "N" THEN 1080 ELSE IF B$ < > "Y" THEN 1190
1210 A$ = "----------------------------------------"
1220 DIM X(250),Y(250),J(500),K(500),AR(500),AINER(500),E(500)
1230 DIM JB(20),K1(50),W(50,2),P(50,2),V1(5,50),VA(5,50)
1240 DIM H1(5,50),HA(5,50),K2(50),PJ(50,3)
1250 IF NH = 0 THEN 1290
1260 Q$ = "F" + C$ : OPEN "O",#2,Q$
1270 DIM AM(500),NHT(4),NT(4),T(4,100),V(4,100),NW(50,2)
1280 DIM NV(5,50),NH(5,50),NP(50,3)
1290 IF F$ = "N" THEN 1320
1300 FOR I = 1 TO NM : E(I) = E : NEXT I
1310 '
1320 ' ** INPUT NODAL POINT COORDINATES **
1330 '
1340 FOR I = 1 TO NJ
1350 PRINT CHR$(12) : PRINT : PRINT A$
1360 PRINT "     INPUT NODAL POINT COORDINATES"
1370 PRINT A$ : PRINT
1380 PRINT "  NODE  NO. ";I: PRINT "  =============="
1390 PRINT : INPUT "  X-COORDINATE.........= ";X(I)
1400 INPUT "  Y-COORDINATE.........= ";Y(I) : PRINT
1410 INPUT " DO YOU ACCEPT THESE DATA ? (Y/N) ";B$
1420 IF B$ = "N" THEN 1350 ELSE IF B$ < > "Y" THEN 1410
1430 PRINT : NEXT I
1440 '
1450 ' ** INPUT NODE NO. & PROPERTIES OF EACH MEMBER **
1460 '
1470 FOR I = 1 TO NM
1480 PRINT CHR$(12) : PRINT : PRINT A$
1490 PRINT "     INPUT NODE NO. & PROP. OF MEMBERS"
1500 PRINT A$ : PRINT
1510 PRINT "  MEMBER NO. ";I : PRINT"  =============="
1520 PRINT : INPUT "  NODE J..............= ";J(I)
1530 INPUT "  NODE K..............= ";K(I)
1540 INPUT "  CROSS SECTIONL AREA..= ";AR(I)
```

```
1550 INPUT "  MOMENT OF INERTIA....= ";AINER(I)
1560 IF NH = 0 THEN 1580
1570 INPUT "  MASS PER UNIT LENGTH.= ";AM(I)
1580 IF F$ = "Y" GOTO 1600
1590 INPUT "  YOUNG'S MODULUS......= ";E(I)
1600 PRINT : INPUT " DO YOU ACCEPT THESE DATA ? (Y/N) ";B$
1610 IF B$ = "N" THEN 1480 ELSE IF B$ < > "Y" THEN 1600
1620 PRINT : NEXT I
1630 '
1640 ' ** INPUT BOUNDARY CONDITIONS AT NODES **
1650 '
1660 PRINT CHR$(12)
1670 PRINT A$ : PRINT " INPUT BC. AT NODES ARE FIXED "
1680 PRINT A$ : PRINT : FOR I = 1 TO JBC
1690 INPUT "  NODE NO. ";JB(I) : NEXT I
1700 PRINT : INPUT " DO YOU ACCEPT THESE DATA ? (Y/N) ";B$
1710 IF B$ = "N" THEN 1660 ELSE IF B$ < > "Y" THEN 1700
1720 '
1730 PRINT #1,NM;NJ;JBC;E;NH;ML;JL
1740 FOR I = 1 TO NJ : PRINT #1,X(I);Y(I) : NEXT I
1750 FOR I = 1 TO NM : PRINT #1,J(I);K(I);AR(I);AINER(I);E(I)
1760 IF NH > 0 THEN PRINT #1,AM(I)
1770 NEXT I
1780 FOR I = 1 TO JBC : PRINT #1,JB(I) : NEXT I
1790 IF NH = 0 THEN 2000
1800 '
1810 ' **  INPUT DATA OF LOAD HISTORY  **
1820 '
1830 FOR J = 1 TO NH
1840 PRINT CHR$(12)
1850 PRINT : PRINT"  ***********************************************"
1860 PRINT"  **  INPUT DATA OF HISTORY LOAD TYPE NO.";J;" **"
1870 PRINT"  ***********************************************" : PRINT
1880 INPUT " IS THIS WIND HISTORY ? (Y/N) ";B$
1890 IF B$ = "N" THEN NHT(J) = 0
1900 IF B$ = "Y" THEN NHT(J) = 1
1910 IF B$ < > "Y" AND B$ < > "N" THEN 1880
1920 INPUT " NO. OF POINT OF THIS FUNCTION ";NT(J)
1930 FOR I = 1 TO NT(J) : PRINT
1940 PRINT " TIME VALUES AT POINT ";I;: INPUT T(J,I)
1950 PRINT " LOAD VALUES AT POINT ";I;: INPUT V(J,I) : NEXT I
1960 PRINT : INPUT " DO YOU ACCEPT THESE DATA ? (Y/N) ";B$
1970 IF B$ = "N" THEN 1840 ELSE IF B$ < > "Y" THEN 1960
1980 PRINT : NEXT J
1990 '
2000 ' INPUT MEMBER LOADS IN LOCAL COORDINATES
2010 '
2020 IF ML = 0 THEN 2380
2030 FOR I = 1 TO ML
2040 PRINT CHR$(12) : PRINT A$
2050 PRINT " INPUT MEMBER LOADS IN GLOBAL COORDINATES"
2060 PRINT A$ : PRINT
2070 INPUT " MEMBER NO. ";K1(I) : PRINT "=================="
2080 INPUT " VERT. UNIF. LODA OR LOAD FACTOR..= ";W(I,1)
2090 IF NH = 0 THEN 2110
```

```
2100 INPUT " HISTORY LODA TYPE NO.............= ";NW(I,1)
2110 INPUT " HORZ. UNIF. LODA OR LOAD FACTOR..= ";W(I,2)
2120 IF NH = 0 THEN 2140
2130 INPUT " HISTORY LODA TYPE NO.............= ";NW(I,2)
2140 PRINT : INPUT " NO. OF VERT. PT. LOAD.............= ";P(I,2)
2150 INPUT " NO. OF HORZ. PT. LOAD.............= ";P(I,1)
2160 PRINT : IF P(I,2) = 0 THEN 2240
2170 FOR J = 1 TO P(I,2) : PRINT " VERT. LOAD NO. ";J
2180 PRINT " ==================="
2190 INPUT " VALUE OF LOAD OR LOAD FACTOR.....= ";V1(J,I)
2200 INPUT " DIST. FROM LEFT..................= ";VA(J,I)
2210 IF NH = 0 THEN 2230
2220 INPUT " HISTORY LOAD TYPE NO.............= ";NV(J,I)
2230 PRINT : NEXT J
2240 IF P(I,1) = 0 THEN 2320
2250 FOR J = 1 TO P(I,1) : PRINT " HORZ. LOAD NO. ";J
2260 PRINT " ==================="
2270 INPUT " VALUE OF LOAD OR LOAD FACTOR.....= ";H1(J,I)
2280 INPUT " DIST. FROM LEFT..................= ";HA(J,I)
2290 IF NH = 0 THEN 2310
2300 INPUT " HISTORY LOAD TYPE NO.............= ";NH(J,I)
2310 PRINT : NEXT J
2320 PRINT : INPUT " DO YOU ACCEPT THESE DATA ? (Y/N) ";B$
2330 IF B$ = "N" THEN 2040 ELSE IF B$ < > "Y" THEN 2320
2340 NEXT I : PRINT
2350 '
2360 ' ** INPUT NODAL LOADS IN GLOBAL COORDINATES **
2370 '
2380 IF JL = 0 THEN 2570
2390 FOR I = 1 TO JL
2400 PRINT CHR$(12) : PRINT A$
2410 PRINT " INPUT NODAL LOADS IN GLOBAL COORDINATES"
2420 PRINT A$ : PRINT
2430 INPUT " NODE NO. ";K2(I) : PRINT "==================="
2440 INPUT " FOREC IN X-DIRECTION.............= ";PJ(I,1)
2450 IF NH = 0 THEN 2470
2460 INPUT " HISTORY LODA TYPE NO.............= ";NP(I,1)
2470 INPUT " FOREC IN Y-DIRECTION.............= ";PJ(I,2)
2480 IF NH = 0 THEN 2500
2490 INPUT " HISTORY LODA TYPE NO.............= ";NP(I,2)
2500 INPUT " MOMENT...........................= ";PJ(I,3)
2510 IF NH = 0 THEN 2530
2520 INPUT " HISTORY LODA TYPE NO.............= ";NP(I,3)
2530 PRINT : INPUT " DO YOU ACCEPT THESE DATA ? (Y/N) ";B$
2540 IF B$ = "N" THEN 2400 ELSE IF B$ < > "Y" THEN 2530
2550 NEXT I : PRINT
2560 '
2570 IF ML = 0 THEN 2690
2580 FOR I = 1 TO ML : PRINT #1,K1(I) : FOR J = 1 TO 2
2590 PRINT #1,W(I,J);P(I,J) : IF NH > 0 THEN PRINT #1,NW(I,J)
2600 NEXT J : IF P(I,2) = 0 THEN 2640
2610 FOR J = 1 TO P(I,2)
2620 PRINT #1,V1(J,I);VA(J,I) : IF NH > 0 THEN PRINT #1,NV(J,I)
2630 NEXT J
```

```
2640 IF P(I,1) = 0 THEN 2680
2650 FOR J = 1 TO P(I,1)
2660 PRINT #1,H1(J,I);HA(J,I) : IF NH > 0 THEN PRINT #1,NH(J,I)
2670 NEXT J
2680 NEXT I
2690 IF JL = 0 THEN 2730
2700 FOR I = 1 TO JL : PRINT #1,K2(I) : FOR J = 1 TO 3
2710 PRINT #1,PJ(I,J) : IF NH > 0 THEN PRINT #1,NP(I,J)
2720 NEXT J,I
2730 IF NH = 0 THEN 2920
2740 '
2750 ' **  INPUT OTHER DATA  **
2760 '
2770 PRINT CHR$(12) : PRINT A$
2780 PRINT "    INPUT OTHER DATA " : PRINT A$ : PRINT
2790 PRINT "  NO. OF FREQUENCIFS TO BE FOUND "
2800 INPUT "      IN THE EIGENVALUE SOLUTION ";NNC : PRINT
2810 PRINT "  DAMPING FACTOR TO BE APPLIDE "
2820 INPUT "                    TO ALL MODE.. ";XI : PRINT
2830 INPUT "  POWER - LAW PARAMETER OF WIND. ";PLP : PRINT
2840 INPUT "  TIME STEP OF INTEGRATION..... ";DT : PRINT
2850 INPUT "  MAX. TIME OF INTEGRATION..... ";TMAX : PRINT
2860 PRINT : INPUT " DO YOU ACCEPT THESE DATA ? (Y/N) ";B$
2870 IF B$ = "N" THEN 2770 ELSE IF B$ < > "Y" THEN 2860
2880 '
2890 PRINT #1,NNC,PLP : PRINT #2,XI,DT,TMAX
2900 FOR J = 1 TO NH : PRINT #1,NHT(J) : PRINT #2,NT(J)
2910 FOR I = 1 TO NT(J) : PRINT #2,T(J,I);V(J,I) : NEXT I,J
2920 CLOSE : CHAIN "OPTION.BAS" : END
```

```
1000 '            ***************************
1010 '            ****     DISPLAY      ****
1020 '            ***************************
1030 '
1040 PRINT CHR$(12)
1050 PRINT "       ************************************
1060 PRINT "       ****    DISPLAY & EDIT DATA    ****
1070 PRINT "       ************************************
1080 'LPT #1
1090 DEFDBL A-R,R-Z : DEFINT I-N : DEFSTR Q
1100 PRINT : INPUT "     FILE NAME ...................";C$
1110 '
1120 '  READ INPUT DATA FROM DISK
1130 '
1140 OPEN "I",#1,C$
1150 INPUT #1,NM,NJ,JBC,E,NH,ML,JL
1160 IF NH < = 1 THEN NN = 6 ELSE NN = NH * 6
1170 NH1 = NN / 6
1180 DIM X(250),Y(250),J(500),K(500),AR(500),AINER(500),E(500)
1190 DIM JB(20),K1(50),W(50,2),P(50,2),V1(5,50),VA(5,50)
1200 DIM H1(5,50),HA(5,50),K2(50),PJ(50,3)
1210 DIM AM(500),NW(50,2),NV(5,50),NH(5,50),NP(50,3)
1220 IF NH = 0 THEN 1240 ELSE Q$ = "F" + C$ : OPEN "I",#2,Q$
1230 DIM NHT(4),NT(4),T(4,50),V(4,50)
1240 FOR I = 1 TO NJ : INPUT #1,X(I),Y(I) : NEXT I
1250 FOR I = 1 TO NM : INPUT #1,J(I),K(I),AR(I),AINER(I),E(I)
1260 IF NH > 0 THEN INPUT #1,AM(I)
1270 NEXT I
1280 FOR I = 1 TO JBC : INPUT #1,JB(I) : NEXT I
1290 IF ML = 0 THEN 1420
1300 FOR I = 1 TO ML : INPUT #1,K1(I) : FOR J = 1 TO 2
1310 INPUT #1,W(I,J),P(I,J)
1320 IF NH > 0 THEN INPUT #1,NW(I,J)
1330 NEXT J : IF P(I,2) = 0 THEN 1370
1340 FOR J = 1 TO P(I,2)
1350 INPUT #1,V1(J,I),VA(J,I) : IF NH > 0 THEN INPUT #1,NV(J,I)
1360 NEXT J
1370 IF P(I,1) = 0 THEN 1410
1380 FOR J = 1 TO P(I,1)
1390 INPUT #1,H1(J,I),HA(J,I) : IF NH > 0 THEN INPUT #1,NH(J,I)
1400 NEXT J
1410 NEXT I
1420 IF JL = 0 THEN 1460
1430 FOR I = 1 TO JL : INPUT #1,K2(I) : FOR J = 1 TO 3
1440 INPUT #1,PJ(I,J) : IF NH > 0 THEN INPUT #1,NP(I,J)
1450 NEXT J,I
1460 IF NH = 0 THEN 1520
1470 INPUT #1,NNC,PLP : FOR J = 1 TO NH : INPUT #1,NHT(J) : NEXT J
1480 IF NH = 0 THEN 1520
1490 INPUT #2,XI,DT,TMAX : FOR J = 1 TO NH
1500 INPUT #2,NT(J) : FOR I = 1 TO NT(J)
1510 INPUT #2,T(J,I),V(J,I) : NEXT I,J
1520 CLOSE
```

```
1530 '
1540 PRINT : PRINT "  NO. OF MEMBER.............= ";NM
1550 PRINT : PRINT "  NO. OF NODES..  ..........= ";NJ
1560 PRINT : PRINT "  NO. OF HISTORY LOAD........ "
1570 PRINT "   [EQ.0 => STATIC ANA.]..... "
1580 PRINT "   [GE.1 => DYANMIC ANA.]...= ";NH
1590 PRINT : PRINT "  NO. OF MEMBERS LOAD.......= ";ML
1600 PRINT : PRINT "  NO. OF JOINTS LOAD........= ";JL
1610 PRINT : PRINT "  NO. OF BOUNDARY NODES.....= ";JBC : PRINT
1620 LOCATE 24,3 : INPUT "PRESS RETURN TO CONTINUE ";Z$
1630 A$  = "####" : B$ = "###.##"
1640 N1 = INT(NJ/15 + 1)
1650 FOR I = 1 TO N1
1660 PRINT CHR$(12):PRINT "   NODE NO. ¦  X - COORDINATE  ¦  Y - COORDINATE
1670 PRINT " ___________¦_________________¦_________________
1680 PRINT "           ¦                 ¦
1690 FOR J = 1 TO 15 : N2 = (I - 1) * 15 + J : IF N2 > NJ THEN 1730
1700 LOCATE 4 + J,4 : PRINT USING A$;N2 : LOCATE 4 + J,13 : PRINT "¦"
1710 LOCATE 4 + J,20 : PRINT USING B$;X(N2) : LOCATE 4 + J,32 : PRINT "¦"
1720 LOCATE 4 + J,39 : PRINT USING B$;Y(N2) : NEXT J
1730 PRINT " ___________¦_________________¦_________________
1740 LOCATE 23,10 : PRINT "RETURN => CONTINUE , G => NEXT DATA SET , E => EXIT"
1750 LOCATE 22,10 : PRINT "WHAT DO YOU EDIT NODE NO.";:INPUT CHO$
1760 IF CHO$ = "" THEN 1870
1770 IF CHO$ = "G" THEN 1880
1780 IF CHO$ = "E" THEN 4250
1790 CHOICE = VAL(CHO$)
1800 IF CHOICE <= (I - 1)* 15 THEN 1750 ELSE IF CHOICE > I * 15 THEN 1750
1810 BUF$ = "      " :LC = 4 + (CHOICE -(I - 1)* 15) : LO = 20 : GOSUB 4600
1820 IF BUF$ = "      " THEN 1840 ELSE X(CHOICE) = VAL(BUF$)
1830 LOCATE LC,LO : PRINT USING B$;X(CHOICE)
1840 BUF$ = "      " :LC = 4 + (CHOICE -(I - 1)* 15) : LO = 39 : GOSUB 4600
1850 IF BUF$ = "      " THEN 1750 ELSE Y(CHOICE) = VAL(BUF$)
1860 LOCATE LC,LO : PRINT USING B$;Y(CHOICE) : GOTO 1750
1870 NEXT I
1880 '
1890 A$ = "###   ¦" : B$ = "##.#### ¦" : C$ = "#.#######¦" : D$ = "#########"
1900 N1 = INT(NM/15 + 1)
1910 FOR I = 1 TO N1
1920 PRINT CHR$(12)
1930 PRINT "  MEM. NO.¦ NODE J ¦ NODE K ¦  AREA  ¦  MASS  ¦ INERTIA ¦ MODULUS
1940 PRINT " _________¦_______¦_______¦_______¦_______¦_________¦________
1950 PRINT "          ¦       ¦       ¦       ¦       ¦         ¦
1960 FOR J = 1 TO 15 : N2 = (I - 1)* 15 + J : IF N2 > NM THEN 2040
1970 LOCATE 4 + J,5 : PRINT USING A$;N2
1980 LOCATE 4 + J,14 : PRINT USING A$;J(N2)
1990 LOCATE 4 + J,23 : PRINT USING A$;K(N2)
2000 LOCATE 4 + J,30 : PRINT USING B$;AR(N2)
2010 LOCATE 4 + J,39 : PRINT USING B$;AM(N2)
2020 LOCATE 4 + J,48 : PRINT USING C$;AINER(N2)
2030 LOCATE 4 + J,58 : PRINT USING D$;E(N2) : NEXT J
2040 PRINT " _________¦_______¦_______¦_______¦_______¦_________¦________
2050 LOCATE 23,10 : PRINT "RETURN => CONTINUE , G => NEXT DATA SET , E => EXIT"
2060 LOCATE 22,10 : PRINT "WHAT DO YOU EDIT MEM. NO.";:INPUT CHO$
```

```
2070 IF CHO$ = "" THEN 2130
2080 CHOICE = VAL(CHO$)
2090 IF CHO$ = "G" THEN 2300
2100 IF CHO$ = "E" THEN 4250
2110 IF CHOICE <= (I - 1)* 15 THEN 2060 ELSE IF CHOICE > I * 15 THEN 2060
2120 GOSUB 2140 : GOTO 2060
2130 NEXT I : GOTO 2300
2140 ' SUBROUTINE EDITOR
2150 LL = 0 : LC = 4 +(CHOICE -(I - 1)* 15)
2160 LL = LL + 1 : IF LL = 1 THEN BUF$ = "   " : LO = 14
2170 IF LL = 2 THEN BUF$ = "   " : LO = 23
2180 IF LL = 3 THEN BUF$ = "       " : LO = 30
2190 IF LL = 4 THEN BUF$ = "       " : LO = 39
2200 IF LL = 5 THEN BUF$ = "         " : LO = 48
2210 IF LL = 6 THEN BUF$ = "         " : LO = 58
2220 GOSUB 4600
2230 IF LL = 1 THEN IF BUF$ = "   " THEN 2160 ELSE J(CHOICE)=VAL(BUF$) : LOCATE
LC,LO : PRINT USING A$;J(CHOICE) : GOTO 2160
2240 IF LL = 2 THEN IF BUF$ = "   " THEN 2160 ELSE K(CHOICE)=VAL(BUF$) : LOCATE
LC,LO : PRINT USING A$;K(CHOICE) : GOTO 2160
2250 IF LL = 3 THEN IF BUF$ = "       " THEN 2160 ELSE AR(CHOICE)=VAL(BUF$) : LO
CATE LC,LO : PRINT USING B$;AR(CHOICE) : GOTO 2160
2260 IF LL = 4 THEN IF BUF$ = "       " THEN 2160 ELSE AM(CHOICE)=VAL(BUF$) : LO
CATE LC,LO : PRINT USING B$;AM(CHOICE) : GOTO 2160
2270 IF LL = 5 THEN IF BUF$ = "         " THEN 2160 ELSE AINER(CHOICE)=VAL(BUF$)
 : LOCATE LC,LO : PRINT USING C$;AINER(CHOICE) : GOTO 2160
2280 IF LL = 6 THEN IF BUF$ = "         " THEN RETURN ELSE E(CHOICE)=VAL(BUF$) :
 LOCATE LC,LO : PRINT USING D$;E(CHOICE) : RETURN
2290 '
2300 A$ = "###" : PRINT CHR$(12)
2310 PRINT "  BOUNDARY CONDITIONS AT NODE NUMBER"
2320 PRINT " ____________________________________"
2330 PRINT "                    |                "
2340 PRINT "    SUPPORT NO.     |    NODE NO.    "
2350 PRINT " ___________________|________________"
2360 PRINT "                    |                "
2370 FOR I = 1 TO JBC
2380 LOCATE 6 + I,10 : PRINT USING A$;I
2390 LOCATE 6 + I,20 : PRINT "|"
2400 LOCATE 6 + I,27 : PRINT USING A$;JB(I) : NEXT I
2410 PRINT " ___________________|________________" : PRINT
2420 LOCATE 23,10 : PRINT "RETURN => CONTINUE , G => NEXT DATA SET , E => EXIT"
2430 LOCATE 22,10 : PRINT "WHAT DO YOU EDIT SUPPORT NO.";:INPUT CHO$
2440 IF CHO$ = "" THEN 2510
2450 IF CHO$ = "G" THEN 2510
2460 IF CHO$ = "E" THEN 4250
2470 CHOICE = VAL(CHO$)
2480 IF CHOICE <= 0 THEN 2430 ELSE IF CHOICE > JBC THEN 2430
2490 BUF$ = "   " : LC = 6 + CHOICE : LO = 27 : GOSUB 4600
2500 IF BUF$ = "   " THEN 2430 ELSE JB(CHOICE)=VAL(BUF$) : LOCATE LC,LO : PRINT
USING A$;JB(CHOICE) : GOTO 2430
2510 IF NH = 0 THEN 2820
2520 A$ = "##      |" : B$ = "##.##     |" : C$ = "#####.###"
2530 FOR I = 1 TO NH : PRINT CHR$(12)
2540 PRINT : PRINT "           HISTORY LOAD TYPE NO. ";I
```

```
2550 IF NHT(I) = 0 THEN 2570
2560 PRINT "        THIS IS WIND LOAD AT 9 m. HIGHT." : GOTO 2580
2570 PRINT "             THIS IS NOT WIND LOAD "
2580 PRINT " ____________________________________________"
2590 PRINT "              |              |   "
2600 PRINT "    POINT NO. |  TIME(SEC.)  |  LOAD VALUE   "
2610 PRINT " _____________|______________|_______________"
2620 PRINT "              |              |     "
2630 FOR J =1 TO NT(I)
2640 LOCATE 8 + J,8 : PRINT USING A$;J
2650 LOCATE 8 + J,21 : PRINT USING B$;T(I,J)
2660 LOCATE 8 + J,34 : PRINT USING C$;V(I,J) : NEXT J
2670 PRINT " _____________|______________|_______________"
2680 LOCATE 23,10 : PRINT "RETURN => CONTINUE , G => NEXT DATA SET , E => EXIT"
2690 LOCATE 22,10 : PRINT "WHAT DO YOU EDIT POINT NO.";:INPUT CHO$
2700 IF CHO$ = "" THEN 2810
2710 IF CHO$ = "G" THEN 2820
2720 IF CHO$ = "E" THEN 4250
2730 CHOICE = VAL(CHO$)
2740 IF CHOICE <= 0 THEN 2690 ELSE IF CHOICE > NT(I) THEN 2690
2750 BUF$ = "     " : LC = 8 + CHOICE : LO = 21 : GOSUB 4600
2760 IF BUF$ = "     " THEN 2780 ELSE T(I,CHOICE) = VAL(BUF$)
2770 LOCATE LC,LO : PRINT USING B$;T(I,CHOICE)
2780 BUF$ = "         " : LC = 8 + CHOICE : LO = 34 : GOSUB 4600
2790 IF BUF$ = "         " THEN 2690 ELSE V(I,CHOICE) = VAL(BUF$)
2800 LOCATE LC,LO : PRINT USING C$;V(I,CHOICE) : GOTO 2690
2810 NEXT I
2820 IF ML = 0 THEN 3460
2830 A$ = "### |" : B$ = "#.###|" : C$ = "##.##|" : D$ = "###"
2840 I1 = INT(ML / 10)+ 1 : M1 = 0 : FOR I = 1 TO I1 : PRINT CHR$(12)
2850 PRINT "         | UNIFORM LOAD ON MEMBER|        POINT LOAD ON MEMBER"
2860 PRINT " _______|_____________________|_____________________________
 "
_
2870 PRINT "    |    |          |          |                |      "
2880 PRINT "LOAD|MEM.|  VERTICAL | HORIZONTAL|    VERTICAL     |    HORIZONTAL"
2890 PRINT "MEM.|NO. |__________|__________|________________|_____________
 "
_
2900 PRINT "NO. |    |    |     |    |     |    |     |    |     |    |"
2910 PRINT "    |    |VALUE|H.NO.|VALUE|H.NO.|VALUE|DIST.|H.NO.|VALUE|DIST.|H.NO
."
2920 PRINT "____|____|_____|_____|_____|_____|_____|_____|_____|_____|_____|____
 "
_
2930 PRINT "    |    |    |     |    |     |    |     |    |     |    |     |"
2940 II = 0 : FOR K = 1 TO 10 : II = II + 1 : M1 = M1 + 1
2950 IF M1 > ML THEN 3100
2960 LOCATE 10 + II,1 : PRINT USING A$;M1
2970 LOCATE 10 + II,6 : PRINT USING A$;K1(M1)
2980 FOR J = 1 TO 2 : JJ = (J - 1)* 12
2990 LOCATE 10 + II,11 + JJ : PRINT USING B$;W(M1,J)
3000 LOCATE 10 + II,18 + JJ : PRINT USING A$;NW(M1,J) : NEXT J
3010 IF P(M1,2) = 0 AND P(M1,1) = 0 THEN NN = 1 : GOTO 3030
3020 NN = P(M1,2) : IF NN < P(M1,1) THEN NN = P(M1,1)
3030 FOR J = 1 TO NN : II = II + J - 1
3040 LOCATE 10 + II,35 : PRINT USING B$;V1(J,M1)
3050 LOCATE 10 + II,41 : PRINT USING C$;VA(J,M1)
```

```
3060 LOCATE 10 + II,48 : PRINT USING A$;NV(J,M1)
3070 LOCATE 10 + II,53 : PRINT USING B$;H1(J,M1)
3080 LOCATE 10 + II,59 : PRINT USING C$;HA(J,M1)
3090 LOCATE 10 + II,66 : PRINT USING D$;NH(J,M1) : NEXT J,K
3100 PRINT "____¦____¦_____¦_____¦_____¦_____¦_____¦_____¦_____¦_____¦_____¦____
 "
3110 LOCATE 23,10 : PRINT "RETURN => CONTINUE , G => NEXT DATA SET , E => EXIT"
3120 LOCATE 22,10 : PRINT "WHAT DO YOU EDIT LOAD MEM. NO.";:INPUT CHO$
3130 IF CHO$ = "" THEN 3190
3140 IF CHO$ = "G" THEN 3460
3150 IF CHO$ = "E" THEN 4250
3160 CHOICE = VAL(CHO$)
3170 IF CHOICE <= (I - 1)* 10 THEN 3120 ELSE IF CHOICE > I * 10 THEN 3120
3180 GOSUB 3200 : GOTO 3120
3190 NEXT I : GOTO 3460
3200 ' SUBROUTINE EDITOR
3210 LL = 0 : LC = 10 +(CHOICE -(I - 1)* 10)
3220 LL = LL + 1 : IF LL = 1 THEN BUF$ = "    " : LO = 6
3230 IF LL = 2 THEN BUF$ = "     " : LO = 11
3240 IF LL = 3 THEN BUF$ = "   " : LO = 18
3250 IF LL = 4 THEN BUF$ = "     " : LO = 23
3260 IF LL = 5 THEN BUF$ = "   " : LO = 30
3270 IF LL = 6 THEN BUF$ = "     " : LO = 35
3280 IF LL = 7 THEN BUF$ = "     " : LO = 41
3290 IF LL = 8 THEN BUF$ = "   " : LO = 48
3300 IF LL = 9 THEN BUF$ = "     " : LO = 53
3310 IF LL = 10 THEN BUF$ = "     " : LO = 59
3320 IF LL = 11 THEN BUF$ = "   " : LO = 66
3330 GOSUB 4600
3340 IF LL = 1 THEN IF BUF$ = "    " THEN 3220 ELSE K1(CHOICE)=VAL(BUF$) : LOCAT
E LC,LO : PRINT USING A$;K1(CHOICE) : GOTO 3220
3350 IF LL = 2 THEN IF BUF$ = "     " THEN 3220 ELSE W(CHOICE,1)=VAL(BUF$) : LOC
ATE LC,LO : PRINT USING B$;W(CHOICE,1) : GOTO 3220
3360 IF LL = 3 THEN IF BUF$ = "   " THEN 3220 ELSE NW(CHOICE,1)=VAL(BUF$) : LOCA
TE LC,LO : PRINT USING A$;NW(CHOICE,1) : GOTO 3220
3370 IF LL = 4 THEN IF BUF$ = "     " THEN 3220 ELSE W(CHOICE,2)=VAL(BUF$) : LOC
ATE LC,LO : PRINT USING B$;W(CHOICE,2) : GOTO 3220
3380 IF LL = 5 THEN IF BUF$ = "   " THEN 3220 ELSE NW(CHOICE,2)=VAL(BUF$) : LOCA
TE LC,LO : PRINT USING A$;NW(CHOICE,2) : GOTO 3220
3390 IF LL = 6 THEN IF BUF$ = "     " THEN 3220 ELSE V1(1,CHOICE)=VAL(BUF$) : LO
CATE LC,LO : PRINT USING B$;V1(1,CHOICE) : GOTO 3220
3400 IF LL = 7 THEN IF BUF$ = "     " THEN 3220 ELSE VA(1,CHOICE)=VAL(BUF$) : LO
CATE LC,LO : PRINT USING C$;VA(1,CHOICE) : GOTO 3220
3410 IF LL = 8 THEN IF BUF$ = "   " THEN 3220 ELSE NV(1,CHOICE)=VAL(BUF$) : LOCA
TE LC,LO : PRINT USING A$;NV(1,CHOICE) : GOTO 3220
3420 IF LL = 9 THEN IF BUF$ = "     " THEN 3220 ELSE H1(1,CHOICE)=VAL(BUF$) : LO
CATE LC,LO : PRINT USING B$;H1(1,CHOICE) : GOTO 3220
3430 IF LL = 10 THEN IF BUF$ = "     " THEN 3220 ELSE HA(1,CHOICE)=VAL(BUF$) : L
OCATE LC,LO : PRINT USING C$;HA(1,CHOICE) : GOTO 3220
3440 IF LL = 11 THEN IF BUF$ = "   " THEN RETURN ELSE NH(1,CHOICE)=VAL(BUF$) : L
OCATE LC,LO : PRINT USING D$;NH(1,CHOICE) : RETURN
3450 '
3460 IF JL = 0 THEN 3960
3470 A$ = "###" : B$ = "¦  ###.###" : C$ = "¦ ###"
3480 II = INT(JL / 10)+ 1 : FOR I = 1 TO II : PRINT CHR$(12)
```

```
3490 PRINT "                    JOINT LOAD   "
3500 PRINT "_____________________________________________________________"
3510 PRINT "     |     |                |                |                |"
3520 PRINT "     |     |  X-COORDINATE  |  Y-COORDINATE  |    ROTATION    |"
3530 PRINT "JOINT|NODE |________________|________________|________________|"
3540 PRINT " LOAD| NO. |       |        |       |        |       |        |"
3550 PRINT " NO. |     |  VALUE  |H.NO. |  VALUE  |H.NO. |  VALUE  |H.NO. |"
3560 PRINT "_____|_____|_________|______|_________|______|_________|______|"
3570 PRINT "     |     |         |      |         |      |         |      |"
3580 J1 = 0 : FOR J = 1 TO 10 : J1 = J1 + 1
3590 IF J1 > JL THEN 3680
3600 LOCATE 10 + J,2 : PRINT USING A$;J1
3610 LOCATE 10 + J,6 : PRINT USING C$;K2(J1)
3620 LOCATE 10 + J,12 : PRINT USING B$;PJ(J1,1)
3630 LOCATE 10 + J,23 : PRINT USING C$;NP(J1,1)
3640 LOCATE 10 + J,29 : PRINT USING B$;PJ(J1,2)
3650 LOCATE 10 + J,40 : PRINT USING C$;NP(J1,2)
3660 LOCATE 10 + J,46 : PRINT USING B$;PJ(J1,3)
3670 LOCATE 10 + J,57 : PRINT USING C$;NP(J1,3) : NEXT J
3680 PRINT " ____|_____|_________|______|_________|______|_________|______"
3690 LOCATE 23,10 : PRINT "RETURN => CONTINUE , G => NEXT DATA SET , E => EXIT"
3700 LOCATE 22,10 : PRINT "WHAT DO YOU EDIT JOINT LOAD NO.";:INPUT CHO$
3710 IF CHO$ = "" THEN 3770
3720 IF CHO$ = "G" THEN 3960
3730 IF CHO$ = "E" THEN 4250
3740 CHOICE = VAL(CHO$)
3750 IF CHOICE <= (I - 1)* 10 THEN 3700 ELSE IF CHOICE > I * 10 THEN 3700
3760 GOSUB 3780 : GOTO 3700
3770 NEXT I : GOTO 3940
3780 ' SUBROUTINE EDITOR
3790 LL = 0 : LC = 10 +(CHOICE -(I - 1)* 10)
3800 LL = LL + 1 : IF LL = 1 THEN BUF$ = "   " : LO = 8
3810 IF LL = 2 THEN BUF$ = "       " : LO = 15
3820 IF LL = 3 THEN BUF$ = "   " : LO = 25
3830 IF LL = 4 THEN BUF$ = "       " : LO = 32
3840 IF LL = 5 THEN BUF$ = "   " : LO = 42
3850 IF LL = 6 THEN BUF$ = "       " : LO = 49
3860 IF LL = 7 THEN BUF$ = "   " : LO = 59
3870 GOSUB 4600
3880 IF LL = 1 THEN IF BUF$ = "   " THEN 3800 ELSE K2(CHOICE)=VAL(BUF$) : LOCATE
 LC,LO : PRINT USING A$;K2(CHOICE) : GOTO 3800
3890 IF LL = 2 THEN IF BUF$ = "       " THEN 3800 ELSE PJ(CHOICE,1)=VAL(BUF$) :
LOCATE LC,12 : PRINT USING B$;PJ(CHOICE,1) : GOTO 3800
3900 IF LL = 3 THEN IF BUF$ = "   " THEN 3800 ELSE NP(CHOICE,1)=VAL(BUF$) : LOCA
TE LC,23 : PRINT USING C$;NP(CHOICE,1) : GOTO 3800
3910 IF LL = 4 THEN IF BUF$ = "       " THEN 3800 ELSE PJ(CHOICE,2)=VAL(BUF$) :
LOCATE LC,29 : PRINT USING B$;PJ(CHOICE,2) : GOTO 3800
3920 IF LL = 5 THEN IF BUF$ = "   " THEN 3800 ELSE NP(CHOICE,2)=VAL(BUF$) : LOCA
TE LC,40 : PRINT USING C$;NP(CHOICE,2) : GOTO 3800
3930 IF LL = 6 THEN IF BUF$ = "       " THEN 3800 ELSE PJ(CHOICE,3)=VAL(BUF$) :
LOCATE LC,46 : PRINT USING B$;PJ(CHOICE,3) : GOTO 3800
3940 IF LL = 7 THEN IF BUF$ = "   " THEN RETURN ELSE NP(CHOICE,3)=VAL(BUF$) : LO
CATE LC,57 : PRINT USING C$;NP(CHOICE,3) : RETURN
3950 '
```

```
3960 IF NH = 0 THEN 4250
3970 PRINT CHR$(12) : PRINT "          OTHER DATA "
3980 PRINT " ____________________________________" : PRINT
3990 PRINT "  NO. OF FREQUENCIFS TO BE FOUND "
4000 PRINT "      IN THE EIGENVALUE SOLUTION = ";NNC : PRINT
4010 PRINT "  DAMPING FACTOR TO BE APPLIDE "
4020 PRINT "                    TO ALL MODE.. = ";XI : PRINT
4030 PRINT "  POWER - LAW PARAMETER OF WIND. = ";PLP : PRINT
4040 PRINT "  TIME STEP OF INTEGRATION..... = ";DT : PRINT
4050 PRINT "  MAX. TIME OF INTEGRATION..... = ";TMAX : PRINT
4060 PRINT " ____________________________________" : PRINT
4070 LOCATE 23,10 : PRINT "RETURN => CONTINUE , G => NEXT DATA SET , E => EXIT"
4080 LOCATE 22,10 : PRINT "WHAT DO YOU EDIT OTHER DATA (Y/N)";:INPUT CHO$
4090 IF CHO$ = "" THEN 4250
4100 IF CHO$ = "G" THEN 4250
4110 IF CHO$ = "E" THEN 4250
4120 IF CHO$ = "Y" THEN 4140
4130 IF CHO$ = "N" THEN 4250 ELSE 4080
4140 LC = 3 : LO = 37 : II = 3
4150 FOR I = 1 TO 5 : BUF$ = "     ": IF I > 2 THEN II = 2
4160 LC = LC + II : GOSUB 4600
4170 ON I GOTO 4180,4190,4200,4210,4220
4180 IF BUF$ = "     " THEN 4230 ELSE NNC = VAL(BUF$) : LOCATE LC,LO : PRINT NNC
: GOTO 4230
4190 IF BUF$ = "     " THEN 4230 ELSE XI = VAL(BUF$) : LOCATE LC,LO : PRINT XI :
GOTO 4230
4200 IF BUF$ = "     " THEN 4230 ELSE PLP = VAL(BUF$) : LOCATE LC,LO : PRINT PLP
: GOTO 4230
4210 IF BUF$ = "     " THEN 4230 ELSE DT = VAL(BUF$) : LOCATE LC,LO : PRINT DT :
GOTO 4230
4220 IF BUF$ = "     " THEN 4230 ELSE TMAX = VAL(BUF$) : LOCATE LC,LO : PRINT TMA
X : GOTO 4230
4230 NEXT I : GOTO 4070
4240 END
4250 PRINT CHR$(12) : PRINT "   NAME OF DATA FILE TO BE SAVED " : PRINT
4260 INPUT "   FILE NAME OF STRUCT.........";C$
4270 OPEN "O",#1,C$ : IF NH = 0 THEN 4290
4280 F$ = "F" + C$ : OPEN "O",#2,F$
4290 PRINT #1,NM;NJ;JBC;E;NH;ML;JL
4300 FOR I = 1 TO NJ : PRINT #1,X(I);Y(I) : NEXT I
4310 FOR I = 1 TO NM : PRINT #1,J(I);K(I);AR(I);AINER(I);E(I)
4320 IF NH > 0 THEN PRINT #1,AM(I)
4330 NEXT I
4340 FOR I = 1 TO JBC : PRINT #1,JB(I) : NEXT I
4350 IF ML = 0 THEN 4480
4360 FOR I = 1 TO ML : PRINT #1,K1(I) : FOR J = 1 TO 2
4370 PRINT #1,W(I,J);P(I,J)
4380 IF NH > 0 THEN PRINT #1,NW(I,J)
4390 NEXT J : IF P(I,2) = 0 THEN 4430
4400 FOR J = 1 TO P(I,2)
4410 PRINT #1,V1(J,I);VA(J,I) : IF NH > 0 THEN PRINT #1,NV(J,I)
4420 NEXT J
4430 IF P(I,1) = 0 THEN 4470
4440 FOR J = 1 TO P(I,1)
4450 PRINT #1,H1(J,I);HA(J,I) : IF NH > 0 THEN PRINT #1,NH(J,I)
```

```
4460 NEXT J
4470 NEXT I
4480 IF JL = 0 THEN 4520
4490 FOR I = 1 TO JL : PRINT #1,K2(I) : FOR J = 1 TO 3
4500 PRINT #1,PJ(I,J) : IF NH > 0 THEN PRINT #1,NP(I,J)
4510 NEXT J,I
4520 IF NH = 0 THEN 4580
4530 PRINT #1,NNC,PLP : FOR J = 1 TO NH : PRINT #1,NHT(J) : NEXT J
4540 IF NH = 0 THEN 4580
4550 PRINT #2,XI,DT,TMAX : FOR J = 1 TO NH
4560 PRINT #2,NT(J) : FOR I = 1 TO NT(J)
4570 PRINT #2,T(J,I),V(J,I) : NEXT I,J
4580 CLOSE
4590 CHAIN "OPTION.BAS" : END
4600 ' SUBROUTINE EDITOR
4610 LOCATE LC,LO,1 : LX = 1
4620 ED$ = INPUT$(1)
4630 IF ED$ = CHR$(13) THEN RETURN
4640 IF LX = 1 THEN 4650 ELSE 4660
4650 LOCATE LC,LO : PRINT BUF$ : LOCATE LC,LO,1
4660 IF ED$ = "0" THEN PRINT ED$;:MID$(BUF$,LX,1)=ED$:LX=LX+1:GOTO 4680
4670 PRINT ED$;: MID$(BUF$,LX,1) = ED$ : LX = LX + 1
4680 XX = CSRLIN : YY = POS(0)
4690 IF LX > LEN(BUF$) THEN ED$ = CHR$(13) : GOTO 4630 ELSE 4620
```

```
1000 '            ***************************
1010 '            ****       STIFF       ****
1020 '            ***************************
1030 '
1040 ' ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
1050 ' ::            = SUBPROGRAM  STIFF =                     ::
1060 ' ::  THIS SUBPROGRAM PERFORMS LOCAL & GLOBAL STIFF-      ::
1070 ' ::  NESS AND MASS MATRIX AND LOAD VECTOR OF EACH        ::
1080 ' ::  MEMBERS. MODIFIED STIFFNESS ADN MASS MATRIX &       ::
1090 ' ::  LOAD VECTOR FOR BOUNDARY CONDITION. SAVE LOCAL      ::
1100 ' ::  STIFFNESS & MASS MATRIX IN FILE 'LOSTIFF',LOCAL     ::
1110 ' ::  LOAD VECTOR IN FILE 'LOFORCE',MODIFIED STIFFNESS    ::
1120 ' ::  IN FILE 'GLOSTIFF',MODIFIED MASS IN FILE            ::
1130 ' ::  'GLOMASS',MODIFIED LOAD IN FILE 'GLOFORCE'.         ::
1140 ' ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
1150 '
1160 DEFDBL A-R,R-Z : DEFINT I-N : DEFSTR Q
1170 PRINT CHR$(12) : LOCATE 12,20
1180 INPUT "FILE NAME OF STRUCTURE.......";C$
1190 '
1200 '  READ INPUT DATA FROM DISK
1210 '
1220 OPEN "I",#1,C$
1230 INPUT #1,NM,NJ,JBC,E,NH,ML,JL
1240 IF NH < = 1 THEN NN = 6 ELSE NN = NH * 6
1250 NH1 = NN / 6
1260 DIM X(250),Y(250),J(500),K(500),AR(500),AINER(500),E(500)
1270 DIM JB(20),K1(50),W(50,2),P(50,2),V1(5,50),VA(5,50)
1280 DIM H1(5,50),HA(5,50),K2(50),PJ(50,3),IQ(2),S(6,6)
1290 DIM R(4,6),GS$(21),F$(31),S$(21),FF$(31)
1300 IF NH = 0 THEN 1330 ELSE Q$ = "F" + C$
1310 DIM AM(500),NHT(4),NW(50,2),NV(5,50),NH(5,50),NP(50,3)
1320 DIM SM(6,6),GM$(21)
1330 FOR I = 1 TO NJ : INPUT #1,X(I),Y(I) : NEXT I
1340 FOR I = 1 TO NM : INPUT #1,J(I),K(I),AR(I),AINER(I),E(I)
1350 IF NH > 0 THEN INPUT #1,AM(I)
1360 NEXT I
1370 FOR I = 1 TO JBC : INPUT #1,JB(I) : NEXT I
1380 IF ML = 0 THEN 1510
1390 FOR I = 1 TO ML : INPUT #1,K1(I) : FOR J = 1 TO 2
1400 INPUT #1,W(I,J),P(I,J)
1410 IF NH > 0 THEN INPUT #1,NW(I,J)
1420 NEXT J : IF P(I,2) = 0 THEN 1460
1430 FOR J = 1 TO P(I,2)
1440 INPUT #1,V1(J,I),VA(J,I) : IF NH > 0 THEN INPUT #1,NV(J,I)
1450 NEXT J
1460 IF P(I,1) = 0 THEN 1500
1470 FOR J = 1 TO P(I,1)
1480 INPUT #1,H1(J,I),HA(J,I) : IF NH > 0 THEN INPUT #1,NH(J,I)
1490 NEXT J
1500 NEXT I
1510 IF JL = 0 THEN 1550
1520 FOR I = 1 TO JL : INPUT #1,K2(I) : FOR J = 1 TO 3
1530 INPUT #1,PJ(I,J) : IF NH > 0 THEN INPUT #1,NP(I,J)
1540 NEXT J,I
```

```
1550 IF NH = 0 THEN 1570
1560 INPUT #1,NNC,PLP : FOR J = 1 TO NH : INPUT #1,NHT(J) : NEXT J
1570 CLOSE #1
1580 '
1590 PRINT CHR$(12) : LOCATE 12,20
1600 PRINT " RUNNING STIFFNESS & MASS"
1610 OPEN "R",#1,"LOSTIFF",168 : OPEN "R",#2,"LOFORCE",NN * 8 + 20
1620 FOR I = 1 TO 21
1630 FIELD #1,(I - 1) * 8 AS G$,8 AS S$(I) : NEXT I
1640 FIELD #2,8 AS D1$,8 AS D2$,2 AS DJ$,2 AS DK$
1650 FOR I = 1 TO NN
1660 FIELD #2,(I - 1) * 8 + 20 AS G$,8 AS FF$(I) : NEXT I
1670 OPEN "R",#3,"GLOSTIFF",168 : IF NH = 0 THEN 1690
1680 OPEN "R",#4,"GLOMASS",168
1690 OPEN "R",#5,"GLOFORCE",NN * 8
1700 FOR I = 1 TO 21 : FIELD #3,(I - 1) * 8 AS G$,8 AS GS$(I)
1710 IF NH = 0 THEN 1730
1720 FIELD #4,(I - 1) * 8 AS G$,8 AS GM$(I)
1730 NEXT I
1740 FOR I = 1 TO NN
1750 FIELD #5,(I - 1) * 8 AS G$,8 AS F$(I) : NEXT I
1760 '
1770 '-----START OF CALCULATION
1780 '
1790 FOR LZ = 1 TO NM
1800 X = X(K(LZ)) - X(J(LZ))
1810 Y = Y(K(LZ)) - Y(J(LZ)) : XY = SQR (X * X + Y * Y)
1820 X1 = X / XY : X2 = - Y / XY : Y1 = - X2 : Y2 = X1
1830 X3 = 0 : Y3 = 0 : Z1 = 0 : Z2 = 0 : Z3 = 1
1840 IQ(1) = J(LZ) : IQ(2) = K(LZ)
1850 FOR I = 1 TO 6 : ST(I) = 0 : FOR J = 1 TO 6
1860 S(I,J) = 0 : IF NH > 0 THEN SM(I,J) = 0
1870 NEXT J,I
1880 '
1890 '  CALCULATE LOCAL LOAD VECTOR
1900 '
1910 FOR I = 1 TO NH1 : FOR J = 1 TO 6 : R(I,J) = 0 : NEXT J,I
1920 IF ML = 0 THEN 2270
1930 FOR I = 1 TO ML : IF K1(I) < > LZ THEN 2250
1940 FOR J = 1 TO 2 : YY = 1
1950 IF J = 1 THEN X = Y1 : Y = Y2 ELSE X = X1 : Y = X2
1960 IF NH = 0 THEN I1 = 1 ELSE I1 = NW(I,J)
1970 IF NH = 0 OR J = 1 THEN 2000
1980 IF NHT(I1) = 0 THEN 2000
1990 YY = (((Y(K(LZ)) + Y(J(LZ))) / 2) / 9) ^ (2 * PLP)
2000 R(I1,1) = R(I1,1) + YY * W(I,J) * X * XY / 2
2010 R(I1,2) = R(I1,2) + YY * W(I,J) * Y * XY / 2
2020 R(I1,3) = R(I1,3) + YY * W(I,J) * Y * XY ^ 2 / 12
2030 R(I1,4) = R(I1,1) : R(I1,5) = R(I1,2)
2040 R(I1,6) = -R(I1,3) : NEXT J
2050 IF P(I,2) = 0 THEN 2150
2060 FOR J = 1 TO P(I,2) : J1 = VA(J,I) : J2 = XY - VA(J,I)
2070 IF NH = 0 THEN I1 = 1 ELSE I1 = NV(J,I)
2080 R(I1,1) = R(I1,1) + V1(J,I) * Y1 * J2 / XY
2090 R(I1,2) = R(I1,2) + V1(J,I) * Y2 * J2 ^ 2 * (3 * J1 + J2) / XY ^ 3
```

```
2100 R(I1,3) = R(I1,3) + V1(J,I) * Y2 * J1 * J2 ^ 2 / XY ^ 2
2110 R(I1,4) = R(I1,4) + V1(J,I) * Y1 * J1 / XY
2120 R(I1,5) = R(I1,5) + V1(J,I) * Y2 * J1 ^ 2 * (3 * J2 + J1) / XY ^ 3
2130 R(I1,6) = R(I1,6) - V1(J,I) * Y2 * J2 * J1 ^ 2 / XY ^ 2
2140 NEXT J
2150 IF P(I,1) = 0 THEN 2250
2160 FOR J = 1 TO P(I,1) : J1 = HA(J,I) : J2 = XY - HA(J,I)
2170 IF NH = 0 THEN I1 = 1 ELSE I1 = NH(J,I)
2180 R(I1,1) = R(I1,1) + H1(J,I) * Y1 * J2 / XY
2190 R(I1,2) = R(I1,2) + H1(J,I) * Y2 * J2 ^ 2 * (3 * J1 + J2) / XY ^ 3
2200 R(I1,3) = R(I1,3) + H1(J,I) * Y2 * J1 * J2 ^ 2 / XY ^ 2
2210 R(I1,4) = R(I1,4) + H1(J,I) * Y1 * J1 / XY
2220 R(I1,5) = R(I1,5) + H1(J,I) * Y2 * J1 ^ 2 * (3 * J2 + J1) / XY ^ 3
2230 R(I1,6) = R(I1,6) - H1(J,I) * Y2 * J2 * J1 ^ 2 / XY ^ 2
2240 NEXT J
2250 NEXT I
2260 '
2270 '  CALCULATE LOCAL STIFFNESS & MASS
2280 '
2290 EI = E(LZ) * AINER(LZ) / XY ^ 3
2300 IF NH > 0 THEN AA = XY * AM(LZ) / 420
2310 S(1,1) = AR(LZ) * E(LZ) / XY : S(1,4) = - S(1,1) : S(4,4) = S(1,1)
2320 IF NH = 0 THEN 2340
2330 SM(1,1) = 140 * AA : SM(1,4) = SM(1,1) / 2 : SM(4,4) = SM(1,1)
2340 S(2,2) = 12 * EI : S(2,3) = 6 * XY * EI : S(2,5) = - S(2,2)
2350 S(2,6) = S(2,3) : S(3,3) = 4 * XY * XY * EI : S(3,5) = - S(2,3)
2360 S(3,6) = 2 * XY * XY * EI : S(5,5) = S(2,2) : S(5,6) = - S(2,3)
2370 S(6,6) = S(3,3) : IF NH = 0 THEN 2420
2380 SM(2,2) = 156 * AA : SM(2,3) = 22 * XY * AA : SM(2,5) = 54 * AA
2390 SM(2,6) = - 13 * XY * AA : SM(3,3) = 4 * XY * XY * AA
2400 SM(3,5) = - SM(2,6) : SM(3,6) = - 3 * XY * XY * AA : SM(5,5) = SM(2,2)
2410 SM(5,6) = - 22 * XY * AA : SM(6,6) = SM(3,3)
2420 FOR I = 1 TO 6 : FOR J = I TO 6 : S(J,I) = S(I,J)
2430 IF NH > 0 THEN SM(J,I) = SM(I,J)
2440 NEXT J,I
2450 '
2460 '  SAVE LOCAL STIFFNESS & MASS INTO DATA FILE ' LOSTIFF '
2470 '  AND SAVE LOCAL LOAD VECTOR INTO DATA FILE ' LOFOREC '
2480 '
2490 II = 0 : FOR I = 1 TO 6 : FOR J = I TO 6 : II = II + 1
2500 LSET S$(II) = MKD$(S(I,J)) : NEXT J,I : PUT #1,LZ
2510 LSET D1$ = MKD$(Y1) : LSET D2$ = MKD$(Y2)
2520 LSET DJ$ = MKI$(J(LZ)) : LSET DK$ = MKI$(K(LZ))
2530 I1 = 0 : FOR I = 1 TO NH1 : FOR J = 1 TO 6
2540 I1 = I1 + 1 : LSET FF$(I1) = MKD$(R(I,J))
2550 NEXT J,I : PUT #2,LZ
2560 IF NH = 0 THEN 2620
2570 II = 0 : FOR I = 1 TO 6 : FOR J = I TO 6 : II = II + 1
2580 LSET S$(II) = MKD$(SM(I,J)) : NEXT J,I : PUT #1,NM + LZ
2590 '
2600 '  MODIFILED LOAD VECTOR
2610 '
2620 FOR I = 1 TO NH1 : R1 = R(I,1) : R2 = R(I,2)
2630 R4 = R(I,4) : R5 = R(I,5)
2640 R(I,1) = R1 * X1 + R2 * X2 : R(I,2) = R1 * Y1 + R2 * Y2
```

```
2650 R(I,4) = R4 * X1 + R5 * X2 : R(I,5) = R4 * Y1 + R5 * Y2
2660 NEXT I
2670 IF JL = 0 THEN 2780
2680 FOR I = 1 TO JL : IF K2(I) < > K(LZ) THEN 2720
2690 FOR J = 1 TO 3 : IF NH = 0 THEN I1 = 1 ELSE I1 = NP(I,J)
2700 II = J + 3 : R(I1,II) = R(I1,II) + PJ(I,J)
2710 K2(I) = -K2(I) : NEXT J
2720 IF K2(I) < > J(LZ) THEN 2760
2730 FOR J = 1 TO 3 : IF NH = 0 THEN I1 = 1 ELSE I1 = NP(I,J)
2740 R(I1,J) = R(I1,J) + PJ(I,J)
2750 K2(I) = -K2(I) : NEXT J
2760 NEXT I
2770 '
2780 '  CALCULATE GLOBAL STIFFNESS & MASS
2790 '
2800 FOR II = 1 TO 6 STEP 3 : FOR I = II TO 6
2810 G1 = S(I,II) : G2 = S(I,II + 1) : G3 = S(I,II + 2)
2820 S(I,II) = G1 * X1 + G2 * X2 + G3 * X3
2830 S(I,II + 1) = G1 * Y1 + G2 * Y2 + G3 * Y3
2840 S(I,II + 2) = G1 * Z1 + G2 * Z2 + G3 * Z3
2850 NEXT I : NEXT II : IF NH = 0 THEN 2920
2860 FOR II = 1 TO 6 STEP 3 : FOR I = II TO 6
2870 G1 = SM(I,II) : G2 = SM(I,II + 1) : G3 = SM(I,II + 2)
2880 SM(I,II) = G1 * X1 + G2 * X2 + G3 * X3
2890 SM(I,II + 1) = G1 * Y1 + G2 * Y2 + G3 * Y3
2900 SM(I,II + 2) = G1 * Z1 + G2 * Z2 + G3 * Z3
2910 NEXT I : NEXT II
2920 FOR II = 3 TO 6 STEP 3 : FOR I = 1 TO II
2930 G1 = S(II -2,I) : G2 = S(II - 1,I) : G3 = S(II,I)
2940 S(II - 2,I) = G1 * X1 + G2 * X2 + G3 * X3
2950 S(II - 1,I) = G1 * Y1 + G2 * Y2 + G3 * Y3
2960 S(II,I) = G1 * Z1 + G2 * Z2 + G3 * Z3
2970 NEXT I : NEXT II : IF NH = 0 THEN 3040
2980 FOR II = 3 TO 6 STEP 3 : FOR I = 1 TO II
2990 G1 = SM(II - 2,I) : G2 = SM(II - 1,I) : G3 = SM(II,I)
3000 SM(II - 2,I) = G1 * X1 + G2 * X2 + G3 * X3
3010 SM(II - 1,I) = G1 * Y1 + G2 * Y2 + G3 * Y3
3020 SM(II,I) = G1 * Z1 + G2 * Z2 + G3 * Z3
3030 NEXT I : NEXT II
3040 FOR J = 1 TO 6 : FOR I = J TO 6 : S(J,I) = S(I,J)
3050 IF NH > 0 THEN SM(J,I) = SM(I,J)
3060 NEXT I,J
3070 IF JBC = 0 GOTO 3270
3080 '
3090 '  MODIFIED BOUNARY CONDITIONS
3100 '
3110 FOR I = 1 TO JBC
3120 IF JB(I) < > J(LZ) GOTO 3140
3130 J1 = 0 : GOSUB 3200
3140 IF JB(I) < > K(LZ) GOTO 3160
3150 J1 = 3 : GOSUB 3200
3160 NEXT I
3170 FOR I = 1 TO 6 : FOR J = I TO 6 : S(J,I) = S(I,J)
3180 IF NH > 0 THEN SM(J,I) = SM(I,J)
3190 NEXT J,I : GOTO 3310
```

```
3200 FOR J = 1 TO 3 : J2 = J1 + J
3210 FOR K = 1 TO J2
3220 S(K,J2) = 0 : IF NH > 0 THEN SM(K,J2) = 0
3230 IF K < J2 GOTO 3250
3240 S(K,J2) = 1 : IF NH > 0 THEN SM(K,J2) = 1
3250 NEXT K : IF J2 = 6 GOTO 3290
3260 FOR K = J2 + 1 TO 6
3270 S(J2,K) = 0 : IF NH > 0 THEN SM(J2,K) = 0
3280 NEXT K
3290 NEXT J : RETURN
3300 '
3310 ' SAVE GLOBAL STIFFNESS INTO DATA FILE ' GLOSTIFF '
3320 ' SAVE GLOBAL MASS INTO DATA FILE ' GLOMASS '
3330 ' AND SAVE GLOBAL LOAD INTO DATA FILE ' GLOMASS '
3340 '
3350 II = 0 : FOR I = 1 TO 6 : FOR J = I TO 6 : II = II + 1
3360 LSET GS$(II) = MKD$(S(I,J)) : IF NH = 0 THEN 3380
3370 LSET GM$(II) = MKD$(SM(I,J))
3380 NEXT J,I : PUT #3,LZ : IF NH > 0 THEN PUT #4,LZ
3390 II = 0 : FOR I = 1 TO NH1 : FOR J = I TO 6
3400 II = II + 1 : LSET F$(II) = MKD$(R(I,J))
3410 NEXT J,I : PUT #5,LZ
3420 NEXT LZ : CLOSE
3430 '
3440 '-----END OF CALCULATION
3450 '
3460 OPEN "O",#1,"DUMY" : PRINT #1,Q$
3470 PRINT #1,NM;NJ;JBC;NH;NH1;NNC
3480 FOR I = 1 TO NM : PRINT #1,J(I);K(I) : NEXT I
3490 FOR I = 1 TO JBC : PRINT #1,JB(I): NEXT I : CLOSE
3500 CHAIN "STRUCT.BAS"
3510 END
```

```
1000 '            ***************************
1010 '            ****      STRUCT       ****
1020 '            ***************************
1030 '
1040 ' ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
1050 ' ::           = SUBPROGRAM  STRUCT =                         ::
1060 ' ::  THIS SUBPROGRAM ASSEMBLES GLOBAL STIFFNESS AND          ::
1070 ' ::  MASS MATRIX OF EACH MEMBERS IN COMPACTED FORM           ::
1080 ' ::  AND ASSEMBLES LOAD VECTOR. MODIFIED STIFFNESS           ::
1090 ' ::  AND MASS MATRIX & SAVE STIFFNESS MATRIX IN FILE         ::
1100 ' ::  'STIFFMAT',MASS MATRIX IN FILE 'MASSMAT'. SAVE          ::
1110 ' ::  LOAD VECTOR IN FILE 'STFORCE'. SET INITIAL              ::
1120 ' ::  MATRIX [X1] OF SUBSPACE ITERATION FOR DYNAMIC           ::
1130 ' ::  ANALYSIS AND SAVE IN FILE 'EIGVEC'.                     ::
1140 ' ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
1150 '
1160 CLEAR : PRINT CHR$(12) : LOCATE 12,20
1170 PRINT " RUNNING STRUCTURE "
1180 DEFDBL A-P,R-Z : DEFINT I-N : DEFSTR Q
1190 DIM NDOF(250,3),J(500),K(500),JB(20)
1200 '
1210 '  READ DATA
1220 '
1230 OPEN "I",#1,"DUMY" : INPUT #1,Q$
1240 INPUT #1,NM,NJ,JBC,NH,NH1,NNC
1250 FOR I = 1 TO NM : INPUT #1,J(I),K(I) : NEXT I
1260 FOR I = 1 TO JBC : INPUT #1,JB(I): NEXT I : CLOSE
1270 '
1280 '  FIND MAXIMUM LEFT & RIGHT MEMBER IN THE JOINT AND
1290 '  MAXIMUM DIFFERRENCE NODE NUMBER IN EACH MEMBER.
1300 '  SET NUMBER DEGREE-OF-FREEDOM.
1310 '
1320 MAXL = 0 : MAXR = 0 : MAXD = 0 : NN = 0
1330 FOR I = 1 TO NJ : MAX1 = 0 : MAX2 = 0
1340 FOR J = 1 TO NM
1350 IF J(J) = I THEN MAX1 = MAX1 + 1
1360 IF K(J) = I THEN MAX2 = MAX2 + 1
1370 NEXT J
1380 IF MAX1 > MAXL THEN MAXL = MAX1
1390 IF MAX2 > MAXR THEN MAXR = MAX2
1400 FOR J = 1 TO JBC
1410 IF I = JB(J) THEN 1430
1420 NEXT J : GOTO 1450
1430 FOR K = 1 TO 3 : NDOF(I,K) = 0
1440 NEXT K : GOTO 1470
1450 FOR K = 1 TO 3 : NN = NN + 1
1460 NDOF(I,K) = NN : NEXT K
1470 NEXT I
1480 FOR I = 1 TO NM
1490 FOR J = 1 TO 3 : FOR K = 1 TO 3 : MAX = 0
1500 MAX = ABS(NDOF(J(I),J) - NDOF(K(I),K))
1510 IF MAX > MAXD THEN MAXD = MAX
1520 NEXT K : NEXT J : NEXT I
1530 IF NH = 0 THEN 1650
```

```
1540 '
1550 '  SET NUMBER OF MODE TO BE FOUND IN THIS EXCUTE
1560 '
1570 IF NNC * 2 < NNC + 8 THEN NC = NNC * 2 ELSE NC = NNC + 8
1580 IF NN < NC THEN NC = NN
1590 '
1600 '  SET BUFFER OF FILE 'EIGVEC'
1610 '
1620 AA = NN / 31
1630 IF AA > 1 THEN NN1 = 31 : NN2 = INT(AA + 1) ELSE NN1 = NN : NN2 = 1
1640 '
1650 DIM MEML(5),MEMR(5),SK(21),COL(31,3),GS$(21)
1660 DIM F$(30),FF(750,3),SF(4,6),MAXA(750)
1670 IF NH = 0 THEN 1690
1680 DIM SM(21),COLM(31,3),W(750),GM$(21),GG$(31),WW(750)
1690 OPEN "R",#1,"GLOSTIFF",168 : IF NH = 0 THEN 1710
1700 OPEN "R",#2,"GLOMASS",168
1710 FOR I = 1 TO 21
1720 FIELD #1,(I - 1) * 8 AS G$,8 AS GS$(I) : NEXT I
1730 IF NH = 0 THEN 1760
1740 FOR I = 1 TO 21
1750 FIELD #2,(I - 1) * 8 AS G$,8 AS GM$(I) : NEXT I
1760 OPEN "O",#3,"STIFFMAT"
1770 IF NH > 0 THEN OPEN "O",#4,"MASSMAT"
1780 MAXC = 0 : MAXA(1) = 1
1790 OPEN "R",#5,"GLOFORCE",NH1 * 48
1800 FOR I = 1 TO NH1 * 6
1810 FIELD #5,(I - 1) * 8 AS G$,8 AS F$(I) : NEXT I
1820 FOR I = 1 TO NN : FOR J = 1 TO NH1
1830 FF(I,J) = 0 : NEXT J,I
1840 '
1850 '  ------START TO ASSEMBLES
1860 '
1870 IN = 0 : IN1 = 0 : FOR IJ = 1 TO NJ
1880 IF NDOF(IJ,1) = 0 AND NDOF(IJ,2) = 0 AND NDOF(IJ,3) = 0 THEN 2820
1890 IL = 0 : IR = 0 : INDEX = 0
1900 '
1910 '  FIND MEMBERS AT ENCH JOINT
1920 '
1930 FOR J = 1 TO NM
1940 IF J(J) = IJ THEN IL = IL + 1 ELSE 1960
1950 MEML(IL) = J : INDEX = INDEX + 1
1960 IF K(J) = IJ THEN IR = IR + 1 ELSE 1980
1970 MEMR(IR) = J : INDEX = INDEX + 1
1980 NEXT J
1990 FOR J = 1 TO 21 : FOR JJ = 1 TO 3
2000 COL(J,JJ) = 0 : IF NH > 0 THEN COLM(J,JJ) = 0
2010 NEXT JJ,J
2020 '
2030 '  READ AND ASSEMBLE GLOBAL STIFFNESS AND MASS MATRIX OF EACH MEMBER
2040 '
2050 FOR II = 1 TO INDEX
2060 IF II < = IL THEN MEM = MEML(II) ELSE MEM = MEMR(II - IL)
2070 GET #1,MEM : GET #5,MEM : IF NH > 0 THEN GET #2,MEM
2080 FOR I = 1 TO 21 : SK(I) = CVD(GS$(I)) : NEXT I
```

```
2090 IN = 0 : FOR I = 1 TO NH1 : FOR J = 1 TO 6
2100 IN = IN + 1 : SF(I,J) = CVD(F$(IN)) : NEXT J,I
2110 IF NH = 0 THEN 2130
2120 FOR I = 1 TO 21 : SM(I) = CVD(GM$(I)) : NEXT I
2130 FOR I = 1 TO 3
2140 IF NDOF(IJ,I) = 0 THEN 2520
2150 ND = NDOF(IJ,I)
2160 IF II < = IL THEN IN = 0 ELSE IN = 3
2170 FOR KK = 1 TO NH1 : K1 = I + IN
2180 FF(ND,KK) = FF(ND,KK) + SF(KK,K1) : NEXT KK
2190 I1 = (14 - I) * (I - 1) / 2 + 1 : I2 = (11 - I) * (I + 2) / 2 + 1
2200 IF II < = IL THEN COL(1,I) = COL(1,I) + SK(I1) : GOTO 2220
2210 COL(1,I) = COL(1,I) + SK(I2)
2220 IF NH = 0 THEN 2250
2230 IF II < = IL THEN COLM(1,I) = COLM(1,I) + SM(I1) : GOTO 2250
2240 COLM(1,I) = COLM(1,I) + SM(I2)
2250 FOR KK = 1 TO 3 : IF KK = I THEN 2360
2260 IF NDOF(IJ,KK) = 0 THEN 2360
2270 IF NDOF(IJ,KK) > NDOF(IJ,I) THEN 2360
2280 ND = NDOF(IJ,I) - NDOF(IJ,KK) + 1
2290 I3 = (13 - KK) * KK / 2 - (6 - I)
2300 I4 = (10 - KK) * (KK + 3) / 2 - (3 - I)
2310 IF II < = IL THEN COL(ND,I) = COL(ND,I) + SK(I3) : GOTO 2330
2320 COL(ND,I) = COL(ND,I) + SK(I4)
2330 IF NH = 0 THEN 2360
2340 IF II < = IL THEN COLM(ND,I) = COLM(ND,I) + SM(I3) : GOTO 2360
2350 COLM(ND,I) = COLM(ND,I) + SM(I4)
2360 NEXT KK
2370 IF II < = IL THEN 2380 ELSE 2450
2380 FOR JJ = 1 TO 3
2390 IF NDOF(K(MEML(II)),JJ) = 0 THEN 2440
2400 IF NDOF(K(MEML(II)),JJ) > NDOF(IJ,I) THEN 2440
2410 ND = NDOF(IJ,I) - NDOF(K(MEML(II)),JJ) + 1 : J1 = I1 + (3 - I) + JJ
2420 COL(ND,I) = COL(ND,I) + SK(J1)
2430 IF NH > 0 THEN COLM(ND,I) = COLM(ND,I) + SM(J1)
2440 NEXT JJ : GOTO 2520
2450 FOR JJ = 1 TO 3
2460 IF NDOF(J(MEMR(II-IL)),JJ) = 0 THEN 2510
2470 IF NDOF(J(MEMR(II-IL)),JJ) > NDOF(IJ,I) THEN 2510
2480 ND = NDOF(IJ,I) - NDOF(J(MEMR(II-IL)),JJ) + 1
2490 J1 = (13 - JJ) * JJ / 2 - (3 - I) : COL(ND,I) = COL(ND,I) + SK(J1)
2500 IF NH > 0 THEN COLM(ND,I) = COLM(ND,I) + SM(J1)
2510 NEXT JJ
2520 NEXT I : NEXT II
2530 '
2540 '  FIND AND CHECK MAXIMUM COLUMN HIGHT
2550 '
2560 FOR I = 1 TO 3 : IF NDOF(IJ,I) = 0 THEN 2810
2570 FOR J = 1 TO MAXD + 1 : JS = MAXD + 2 - J
2580 IF COL(JS,I) < > 0 THEN 2600
2590 NEXT J
2600 IF NH = 0 THEN 2650
2610 FOR J = 1 TO MAXD + 1 : JM = MAXD + 2 - J
2620 IF COLM(JM,I) < > 0 THEN 2640
2630 NEXT J
```

```
2640 IF JS = JM THEN 2650 ELSE PRINT " COLUMN HIGHT IS NOT EQUAL" : STOP
2650 IF JS > MAXC THEN MAXC = JS
2660 '
2670 '  SET POSITION OF DIAGONAL OF STIFFNESS AND MASS MATRIX
2680 '
2690 IN1 = IN1 + 1 : MAXA(IN1 + 1) = MAXA(IN1) + JS
2700 '
2710 '  SAVE STIFFNESS INTO DATA FILE 'STIFFMAT'
2720 '
2730 FOR J = 1 TO JS : PRINT #3,COL(J,I) : NEXT J
2740 IF NH = 0 THEN 2810
2750 '
2760 '  SAVE MASS INTO DATA FILE 'MASSMAT'
2770 '
2780 FOR J = 1 TO JS : PRINT #4,COLM(J,I) : NEXT J
2790 '
2800 W(IN1) = COLM(1,I) / COL(1,I) : WW(IN1) = COLM(1,I)
2810 NEXT I
2820 NEXT IJ : CLOSE : IF NH = 0 THEN 3200
2830 '
2840 '  ------END TO ASSEMBLES
2850 '
2860 '  SET INITIAL MATRIX [X1] OF SUBSPACE ITERATION AND SAVE IN DISK
2870 '
2880 OPEN "R",#1,"EIGVEC",NN1 * 8
2890 FOR I = 1 TO NN1 : FIELD #1,(I - 1) * 8 AS G$,8 AS GG$(I) : NEXT I
2900 IN1 = 0 : FOR I = 1 TO NN2 : FOR J = 1 TO NN1
2910 IN1 = IN1 + 1 : IF IN1 > NN THEN 2930
2920 LSET GG$(J) = MKD$(WW(IN1)) : NEXT J
2930 PUT #1,I : NEXT I
2940 FOR J = 1 TO NC - 1 : WW = 0 : IN = 0
2950 FOR I = 1 TO NN : IF W(I) > = WW THEN WW = W(I) : IJ = I
2960 NEXT I : W(IJ) = 0 : FOR KK = 1 TO NN2
2970 FOR K = 1 TO NN1 : LSET GG$(K) = MKD$(0) : NEXT K
2980 IF IN = 1 THEN 3010
2990 K1 = KK * NN1 : IF K1 > = IJ THEN 3000 ELSE 3010
3000 K2 = IJ - (KK - 1) * NN1 : LSET GG$(K2) = MKD$(1) : IN = 1
3010 PUT #1,J * NN2 + KK : NEXT KK,J
3020 '
3030 '  SAVE LOAD VECTOR INTO DATA FILE 'STFORCE'
3040 '
3050 OPEN "O",#2,"STFORCE" : FOR I = 1 TO NH1
3060 FOR J = 1 TO NN : PRINT #2,FF(J,I) : NEXT J,I
3070 '
3080 '  SAVE NUMBER DEGREE-OF-FREEDOM INTO DATA 'NDOF'
3090 '
3100 OPEN "O",#3,"NDOF" : PRINT #3,NM;NJ
3110 FOR I = 1 TO NJ : FOR J = 1 TO 3
3120 PRINT #3,NDOF(I,J) : NEXT J,I : CLOSE
3130 '
3140 OPEN "O",#1,"DUMY1" : PRINT #1,NH
3150 OPEN "O",#2,"DUMY" : PRINT #2,Q$
3160 PRINT #2,NN;MAXC;NNC;NC;NN1;NN2;NH;NH1
3170 FOR I = 1 TO NN + 1 : PRINT #2,MAXA(I) : NEXT I
3180 CLOSE : GOTO 3270
```

```
3190 '
3200 OPEN "O",#1,"DUMY1" : PRINT #1,NH
3210 OPEN "O",#2,"DUMY"
3220 PRINT #2,NN;MAXC;NJ;NM;NH1
3230 FOR I = 1 TO NJ : FOR J = 1 TO 3
3240 PRINT #2,NDOF(I,J) : NEXT J,I
3250 FOR I = 1 TO NN : PRINT #2,FF(I,1);MAXA(I) : NEXT I
3260 PRINT #2,MAXA(NN + 1) : CLOSE
3270 CHAIN "DECOM.BAS"
3280 END
```

```
1000 '            ************************
1010 '            ****     DECOMPOSE    ****
1020 '            ************************
1030 '
1040 ' ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
1050 ' ::            = SUBPROGRAM  DECOM =                    ::
1060 ' ::  THIS SUBPROGRAM CALCULATES (L)*(D)*(L)(T)          ::
1070 ' ::  FACTORIZATION OF STIFFNESS MATRIX. REDUCE AND      ::
1080 ' ::  BACK-SUBSTITUTE DISPLACEMENT VECTOR FOR STATIC     ::
1090 ' ::  ANALYSIS.                                          ::
1100 ' ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
1110 '
1120 CLEAR : PRINT CHR$(12) : LOCATE 12,20
1130 PRINT " RUNING DECOMPOSE "
1140 DEFDBL A-P,R-Z : DEFINT I-N : DEFSTR Q
1150 OPEN "I",#1,"DUMY1" : INPUT #1,NH
1160 DIM MAXA(750) : IF NH = 0 THEN 1250
1170 '
1180 '  READ DATA
1190 '
1200 OPEN "I",#2,"DUMY" : INPUT #2,Q$
1210 INPUT #2,NN,MAXC,NNC,NC,NN1,NN2,NH,NH1
1220 FOR I = 1 TO NN + 1 : INPUT #2,MAXA(I) : NEXT I
1230 CLOSE : GOTO 1320
1240 '
1250 DIM NDOF(250,3),FF(750,1) : OPEN "I",#2,"DUMY"
1260 INPUT #2,NN,MAXC,NJ,NM,NH1
1270 FOR I = 1 TO NJ : FOR J = 1 TO 3
1280 INPUT #2,NDOF(I,J) : NEXT J,I
1290 FOR I = 1 TO NN : INPUT #2,FF(I,1),MAXA(I) : NEXT I
1300 INPUT #2,MAXA(NN + 1) : CLOSE
1310 '
1320 '  ------STRAT TO DECOMPOSE OF STIFFNESS MATRIX
1330 '
1340 '  TO CALCULATE (L)*(D)*(L(T)) FACTORZATION OF STIFFNESS MATRIX
1350 '
1360 IF NN < 2 * MAXC THEN MM = NN ELSE MM = 2 * MAXC
1370 DIM A(2000),DE$(31)
1380 OPEN "I",#1,"STIFFMAT" : J1 = 0 : J2 = 0 : J3 = MM
1390 M1 = MM : JJ = MAXA(M1 + 1) - MAXA(1)
1400 FOR J = 1 TO JJ : J1 = J1 + 1 : INPUT #1,A(J1) : NEXT J
1410 OPEN "R",#2,"DESTIFF",MAXC * 8 : FOR I = 1 TO MAXC
1420 FIELD #2,(I - 1) * 8 AS G$,8 AS DE$(I) : NEXT I
1430 IN2 = 0 : FOR N = 1 TO NN : KN = MAXA(N) - J2
1440 KL = KN + 1 : KU = MAXA(N + 1) - J2 - 1
1450 KH = KU - KL
1460 IF KH < 0 THEN 1630 ELSE IF KH = 0 THEN 1570
1470 K = N - KH : IC = 0 : KT = KU
1480 FOR J = 1 TO KH : IC = IC + 1 : KT = KT - 1
1490 KI = MAXA(K) - J2 : ND = MAXA(K + 1) - KI - J2 - 1
1500 IF ND < = 0 THEN 1560
1510 IF ND > = IC THEN KK = IC ELSE KK = ND
1520 C = 0
```

```
1530 FOR L = 1 TO KK
1540 C = C + A(KI + L) * A(KT + L) : NEXT L
1550 A(KT) = A(KT) - C
1560 K = K + 1 : NEXT J
1570 K = N : B = 0
1580 FOR KK = KL TO KU
1590 K = K - 1 : KI = MAXA(K) - J2 : C = A(KK) / A(KI)
1600 ' IF ABS (C) < 10000000# THEN 1310 ELSE STOP
1610 B = B + C * A(KK) : A(KK) = C : NEXT KK
1620 A(KN) = A(KN) - B
1630 IF A(KN) > 0 THEN 1650
1640 ' IF ISH = 0 THEN STOP ELSE 1270
1650 IF N < J3 THEN 1750 ELSE IF N = NN THEN J1 = 0 : GOTO 1760
1660 IF J3 + MAXC + 1 > NN THEN J3 = NN ELSE J3 = J3 + MAXC + 1
1670 J4 = 0 : FOR M = 1 TO MAXC + 1 : IN2 = IN2 + 1
1680 JJ = MAXA(IN2 + 1) - MAXA(IN2) : J1 = J1 - JJ
1690 FOR I = 1 TO MAXC : LSET DE$(I) = MKD$(0) : NEXT I
1700 FOR J = 1 TO JJ : LSET DE$(J) = MKD$(A(J4 + J)) : NEXT J
1710 J2 = J2 + JJ : J4 = J4 + JJ : PUT #2,IN2 : NEXT M
1720 FOR I = 1 TO J1 : A(I) = A(I + J4) : NEXT I
1730 JJ = MAXA(J3 + 1) - MAXA(M1 + 1) : M1 = J3
1740 FOR I = 1 TO JJ : J1 = J1 + 1 : INPUT #1,A(J1) : NEXT I
1750 NEXT N
1760 FOR I = 1 TO NN - IN2 : IN2 = IN2 + 1
1770 JJ = MAXA(IN2 + 1) - MAXA(IN2)
1780 FOR II = 1 TO MAXC : LSET DE$(II) = MKD$(0) : NEXT II
1790 FOR J = 1 TO JJ : J1 = J1 + 1 : LSET DE$(J) = MKD$(A(J1)) : NEXT J
1800 PUT # 2,IN2 : NEXT I
1810 IF NH = 0 THEN 1860 ELSE CLOSE
1820 OPEN "O",#1,"DUMY" : PRINT #1,Q$
1830 PRINT #1,NN;MAXC;NNC;NC;NN1;NN2;NH1
1840 FOR I = 1 TO NN + 1 : PRINT #1,MAXA(I) : NEXT I
1850 CLOSE : CHAIN"SSPACE.BAS"
1860 '
1870 '  TO REDUCE AND BACK-SUBSITITUTE DISPLACEMENT VECTORS
1880 '
1890 PRINT "  REDUCE AND BACK-SUBSITITUTE " : PRINT
1900 J1 = 0 : J2 = 0 : J3 = 0 : J4 = MM : M1 = MM
1910 I1 = 1 : I2 = MAXC + 1 : FOR II = 1 TO MM : J3 = J3 + 1
1920 GET #2,J3 : JJ = MAXA(II + 1) - MAXA(II)
1930 FOR I = 1 TO JJ : J1 = J1 + 1 : A(J1) = CVD(DE$(I)) : NEXT I,II
1940 FOR L = 1 TO NN : IF L < = M1 THEN 2020
1950 IF M1 + MAXC + 1 < NN THEN J4 = M1 + MAXC + 1 ELSE J4 = NN
1960 JJ = MAXA(I2 + 1) - MAXA(I1) : J1 = J1 - JJ : J2 = J2 + JJ
1970 FOR II = 1 TO J1 : A(II) = A(II + JJ) : NEXT II
1980 FOR I = M1 + 1 TO J4 : J3 = J3 + 1 : GET #2,J3
1990 JJ = MAXA(I + 1) - MAXA(I)
2000 FOR II = 1 TO JJ : J1 = J1 + 1 : A(J1) = CVD(DE$(II)) : NEXT II,I
2010 M1 = J4 : I1 = I2 + 1 : I2 = I1 + MAXC
2020 KL = MAXA(L) - J2 + 1 : KU = MAXA(L + 1) - J2 - 1
2030 IF KU - KL < 0 THEN 2090
2040 K = L : C = 0
2050 FOR KK = KL TO KU : K = K - 1 : C = C + A(KK) * FF(K,1) : NEXT KK
2060 FF(L,1) = FF(L,1) - C : IF L < M1 THEN 2090
```

```
2070 FOR I = I1 TO I2 : IF I > NN THEN 2090
2080 FF(I,1) = FF(I,1) / A(MAXA(I) - J2) : NEXT I
2090 NEXT L
2100 FOR N = I2 + 1 TO NN
2110 FF(N,1) = FF(N,1) / A(MAXA(N) - J2) : NEXT N
2120 IF NN = 1 THEN 2270 ELSE N = NN
2130 FOR L = 2 TO NN : IF N > = I1 THEN 2220
2140 I2 = I1 - MAXC -1 : J3 = MAXC * MAXC * 2
2150 JJ = MAXA(I1) - MAXA(I2) : J2 = J2 - JJ
2160 FOR I = 1 TO 2 * MAXC * MAXC - JJ
2170 A(J3) = A(J3 - JJ) : J3 = J3 - 1 : NEXT I
2180 J1 = 0 : FOR I = I2 TO I1 - 1 : GET #2,I
2190 JJ = MAXA(I + 1) - MAXA(I)
2200 FOR II = 1 TO JJ : J1 = J1 + 1 : A(J1) = CVD(DE$(II))
2210 NEXT II,I : I1 = I2
2220 KL = MAXA(N) - J2 + 1 : KU = MAXA(N + 1) - J2 - 1
2230 IF KU - KL < 0 THEN 2260 ELSE K = N
2240 FOR KK = KL TO KU
2250 K = K - 1 : FF(K,1) = FF(K,1) - A(KK) * FF(N,1) : NEXT KK
2260 N = N - 1 : NEXT L
2270 CLOSE
2280 OPEN "O",#1,"DUMY"
2290 PRINT #1,NN;NJ;NM;NC;NH1
2300 FOR I = 1 TO NJ : FOR J = 1 TO 3
2310 PRINT #1,NDOF(I,J) : NEXT J,I
2320 FOR I = 1 TO NN : PRINT #1,FF(I,1) : NEXT I : CLOSE
2330 CHAIN "S-OUT.BAS"
2340 END
```

```
1000 '            **************************
1010 '            ****     SUBSPACE     ****
1020 '            **************************
1030 '
1040 ' ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
1050 ' ::            = SUBPROGRAM  SSPACE =                        ::
1060 ' ::  THIS SUBPROGRAM TO SOLVE FOR THE SMALLEST               ::
1070 ' ::  EIGENVALUES AND CORRESPONDING EIGENVECTORS THE          ::
1080 ' ::  GENERALIZED EIGENPROBLEM USING THE SUBSPACE             ::
1090 ' ::  ITERATION METHOD. SAVE EIGENVECTORS IN FILE             ::
1100 ' ::  'EIGVEC'.                                              ::
1110 ' ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
1120 '
1130 CLEAR : PRINT CHR$(12) : LOCATE 12,20
1140 PRINT "SUBSPACE ITERRATION METHOD"
1150 '
1160 DEFDBL A-P,R-Z : DEFINT I-N : DEFSTR Q
1170 '
1180 '  READ DATA
1190 '
1200 DIM MAXA(750) : OPEN "I",#1,"DUMY" : INPUT #1,Q$
1210 INPUT #1,NN,MAXC,NNC,NC,NN1,NN2,NH1
1220 FOR I = 1 TO NN + 1 : INPUT #1,MAXA(I) : NEXT I : CLOSE
1230 '
1240 '  INITIALIZATION
1250 '
1260 NCC = NC * (NC + 1) / 2 : LP = 0 : NITEM = 16 : ISH = 0
1270 RT = .000001# : IV = 0 : NSMAX = 12 : N1 = NC + 1
1280 DIM AR(500),BR(500),EIGV(31),VEC(31,31),D(31),TT(31)
1290 DIM W(750),COL(31),DI(750),GM$(31),A(1500),DE$(21)
1300 OPEN "R",#2,"DESTIFF",MAXC * 8
1310 FOR I = 1 TO MAXC
1320 FIELD #2,(I - 1) * 8 AS G$,8 AS DE$(I) : NEXT I
1330 OPEN "R",#4,"EIGVEC",NN1 * 8 : FOR I = 1 TO NN1
1340 FIELD #4,(I - 1) * 8 AS G$,8 AS GM$(I) : NEXT I
1350 '
1360 '  ------START OF ITERATION LOOP
1370 '
1380 IT = 0
1390 IT = IT + 1
1400 '
1410 '  CALCULATE THE PROJECTION OF K AND M
1420 '
1430 IJ = 0
1440 IF NN < 2 * MAXC THEN MM = NN ELSE MM = 2 * MAXC
1450 J = 1
1460 FOR J = 1 TO NC : IN2 = 0 : FOR J2 = 1 TO NN2
1470 GET #4,(J - 1) * NN2 + J2
1480 FOR J3 = 1 TO NN1 : IN2 = IN2 + 1
1490 IF IN2 > NN THEN 1530
1500 W(IN2) = CVD(GM$(J3))
1510 NEXT J3,J2
```

```
1520 '
1530 '  REDUCE AND BACK-SUBSTITUTE ITERATION VECTORS
1540 '
1550 J1 = 0 : J2 = 0 : J3 = 0 : J4 = MM : M1 = MM
1560 I1 = 1 : I2 = MAXC + 1 : FOR II = 1 TO MM : J3 = J3 + 1
1570 GET #2,J3 : JJ = MAXA(II + 1) - MAXA(II)
1580 FOR I = 1 TO JJ : J1 = J1 + 1 : A(J1) = CVD(DE$(I)) : NEXT I,II
1590 FOR L = 1 TO NN : IF L < = M1 THEN 1670
1600 IF M1 + MAXC + 1 < NN THEN J4 = M1 + MAXC + 1 ELSE J4 = NN
1610 JJ = MAXA(I2 + 1) - MAXA(I1) : J1 = J1 - JJ : J2 = J2 + JJ
1620 FOR II = 1 TO J1 : A(II) = A(II + JJ) : NEXT II
1630 FOR I = M1 + 1 TO J4 : J3 = J3 + 1 : GET #2,J3
1640 JJ = MAXA(I + 1) - MAXA(I)
1650 FOR II = 1 TO JJ : J1 = J1 + 1 : A(J1) = CVD(DE$(II)) : NEXT II,I
1660 M1 = J4 : I1 = I2 + 1 : I2 = I1 + MAXC
1670 KL = MAXA(L) - J2 + 1 : KU = MAXA(L + 1) - J2 - 1
1680 IF KU - KL < 0 THEN 1740
1690 K = L : C = 0
1700 FOR KK = KL TO KU : K = K - 1 : C = C + A(KK) * W(K) : NEXT KK
1710 W(L) = W(L) - C : IF L < M1 THEN 1740
1720 FOR I = I1 TO I2 : IF I > NN THEN 1740
1730 W(I) = W(I) / A(MAXA(I) - J2) : NEXT I
1740 NEXT L
1750 FOR N = I2 + 1 TO NN
1760 W(N) = W(N) / A(MAXA(N) - J2) : NEXT N
1770 IF NN = 1 THEN 1920 ELSE N = NN
1780 FOR L = 2 TO NN : IF N > = I1 THEN 1870
1790 I2 = I1 - MAXC -1 : J3 = MAXC * MAXC * 2
1800 JJ = MAXA(I1) - MAXA(I2) : J2 = J2 - JJ
1810 FOR I = 1 TO 2 * MAXC * MAXC - JJ
1820 A(J3) = A(J3 - JJ) : J3 = J3 - 1 : NEXT I
1830 J1 = 0 : FOR I = I2 TO I1 - 1 : GET #2,I
1840 JJ = MAXA(I + 1) - MAXA(I)
1850 FOR II = 1 TO JJ : J1 = J1 + 1 : A(J1) = CVD(DE$(II))
1860 NEXT II,I : I1 = I2
1870 KL = MAXA(N) - J2 + 1 : KU = MAXA(N + 1) - J2 - 1
1880 IF KU - KL < 0 THEN 1910 ELSE K = N
1890 FOR KK = KL TO KU
1900 K = K - 1 : W(K) = W(K) - A(KK) * W(N) : NEXT KK
1910 N = N - 1 : NEXT L
1920 '
1930 FOR I = J TO NC : IN2 = 0 : TA = 0 : FOR J2 = 1 TO NN2
1940 GET #4,(I - 1) * NN2 + J2
1950 FOR J3 = 1 TO NN1 : IN2 = IN2 + 1 : IF IN2 > NN THEN 1970
1960 TA = TA + CVD(GM$(J3)) * W(IN2) : NEXT J3
1970 NEXT J2 : IJ = IJ + 1 : AR(IJ) = TA : NEXT I
1980 IN2 = 0 : FOR J2 = 1 TO NN2 : FOR J3 = 1 TO NN1
1990 IN2 = IN2 + 1 : IF IN2 > NN THEN 2010
2000 LSET GM$(J3) = MKD$(W(IN2)) : NEXT J3
2010 PUT #4,(J - 1) * NN2 + J2 : NEXT J2,J
2020 IJ = 0
2030 FOR J = 1 TO NC
```

```
2040 '
2050 '  EVALUATE PRODUCT OF MASS MATRIX TIMES ITERATION VECTORS
2060 '  AND STORE RESULT IN VECTOR W
2070 '
2080 OPEN "I",#3,"MASSMAT"
2090 IF 2 * MAXC > NN THEN IN1 = NN ELSE IN1 = 2 * MAXC
2100 J1 = 0 : J2 = 1 : J3 = IN1 + 1
2110 JJ = MAXA(J3) - MAXA(J2)
2120 FOR I = 1 TO JJ : J1 = J1 + 1 : INPUT #3,A(J1) : NEXT I
2130 FOR I = 1 TO NN : W(I) = 0 : NEXT I : IN2 = 0
2140 FOR I = 1 TO NN2 : GET #4,(J - 1) * NN2 + I
2150 FOR I1 = 1 TO NN1 : IN2 = IN2 + 1 : IF IN2 > NN THEN 2250
2160 IF IN2 < J3 THEN 2210
2170 J2 = J3 : J4 = 2 * MAXC
2180 IF J3 + J4 < = NN + 1 THEN J3 = J3 + J4 ELSE J3 = NN + 1
2190 JJ = MAXA(J3) - MAXA(J2) : J1 = 0
2200 FOR I2 = 1 TO JJ : J1 = J1 + 1 : INPUT #3,A(J1) : NEXT I2
2210 KL = MAXA(IN2) : KU = MAXA(IN2 + 1) - 1 : KH = MAXA(J2) - 1
2220 II = IN2 + 1 : C = CVD(GM$(I1))
2230 FOR KK = KL TO KU : II = II - 1
2240 W(II) = W(II) + A(KK - KH) * C : NEXT KK,I1,I
2250 CLOSE #3 : IF NN = 1 THEN 2480
2260 OPEN "I",#3,"MASSMAT" : J1 = 0 : J2 = 1 : J3 = IN1 + 1
2270 JJ = MAXA(J3) - MAXA(J2)
2280 FOR I = 1 TO JJ : J1 = J1 + 1 : INPUT #3,A(J1) : NEXT I
2290 IN2 = 0 : FOR I = 1 TO NN2 : GET #4,(J - 1) * NN2 + I
2300 FOR I1 = 1 TO NN1 : IN2 = IN2 + 1 : IF IN2 > NN THEN 2460
2310 IF IN2 < J3 THEN 2360
2320 J2 = J3 : J4 = 2 * MAXC
2330 IF J3 + J4 < = NN + 1 THEN J3 = J3 + J4 ELSE J3 = NN + 1
2340 JJ = MAXA(J3) - MAXA(J2) : J1 = 0
2350 FOR I2 = 1 TO JJ : J1 = J1 + 1 : INPUT #3,A(J1) : NEXT I2
2360 IF IN2 = 1 THEN 2430
2370 KL = MAXA(IN2) + 1 : KU = MAXA(IN2 + 1) - 1 : KH = MAXA(J2) - 1
2380 IF KU - KL < 0 THEN 2430 ELSE II = I1 : AA = 0
2390 FOR KK = KL TO KU : II = II - 1 : IF II < 1 THEN 2410
2400 AA = AA + A(KK - KH) * CVD(GM$(II)) : GOTO 2420
2410 AA = AA + A(KK - KH) * COL(MAXC - II)
2420 NEXT KK : W(IN2) = W(IN2) + AA
2430 NEXT I1
2440 FOR KK = 1 TO MAXC : COL(KK) = CVD(GM$(NN1 - MAXC + KK))
2450 NEXT KK,I
2460 CLOSE #3
2470 '
2480 FOR I = J TO NC : IN2 = 0 : TB = 0 : FOR J2 = 1 TO NN2
2490 GET #4,(I - 1) * NN2 + J2
2500 FOR J3 = 1 TO NN1 : IN2 = IN2 + 1 : IF IN2 > NN THEN 2520
2510 TB = TB + CVD(GM$(J3)) * W(IN2) : NEXT J3,J2
2520 IJ = IJ + 1 : BR(IJ) = TB : NEXT I
2530 IF IV > 0 THEN 2580
2540 IN2 = 0 : FOR J2 = 1 TO NN2 : FOR J3 = 1 TO NN1
2550 IN2 = IN2 + 1 : IF IN2 > NN THEN 2570
2560 LSET GM$(J3) = MKD$(W(IN2)) : NEXT J3
2570 PUT #4,(J - 1) * NN2 + J2 : NEXT J2
2580 NEXT J
```

```
2590 '
2600 '  SOLVE FOR EIGENSYSTEM OF SUBSPACE OPERATORS
2610 '
2620 IF LP = 0 THEN 2760
2630 IND = 1
2640 PRINT "PROJECTION OF A (MATRIX AR)"
2650 II = 1
2660 FOR I = 1 TO NC : TEMP = II + NC - I
2670 FOR J = II TO TEMP : PRINT "AR(";J;") = ";AR(J) : NEXT J
2680 II = II + N1 - I : NEXT I
2690 PRINT "PROJECTION OF B (MATRIX BR)"
2700 II = 1
2710 FOR I = 1 TO NC : TEMP = II + NC - I
2720 FOR J = II TO TEMP : PRINT "BR(";J;") = ";BR(J) : NEXT J
2730 II = II + N1 - I : NEXT I
2740 IF IND = 2 THEN 4070
2750 '
2760 '  SOLVE TO GENERALIZED EIGENPROBLEM USING THE
2770 '        GENERALIZED JACOBI ITERATION
2780 '
2790 '  INITIALIZE EIGENVALUE AND EIGENVECTOR MATRICES
2800 '
2810 N1 = NC + 1 : II = 1
2820 FOR I = 1 TO NC
2830 IF AR(II) < = 0 OR BR(II) < = 0 THEN STOP
2840 TT(I) = AR(II) / BR(II) : EIGV(I) = TT(I) : II = II + N1 - I : NEXT I
2850 FOR I = 1 TO NC : FOR J = 1 TO NC
2860 VEC(I,J) = 0 : NEXT J : VEC(I,I) = 1 : NEXT I
2870 IF NC = 1 THEN 4020
2880 '
2890 '  INITIALIZE SWEEP COUNTER AND BEGIN ITERATION
2900 '
2910 NSWEEP = 0 : NR = NC - 1
2920 NSWEEP = NSWEEP + 1
2930 '
2940 '  CHECK IF PRESENT OFF - DIAGONAL ELEMENT IS LARGE ENOUGH
2950 '  TO REQUIRE ZEROING
2960 '
2970 EPS = (.01 ^ NSWEEP) ^ 2
2980 FOR J = 1 TO NR : JP1 = J + 1
2990 JM1 = J - 1 : LJK = JM1 * NC - JM1 * J / 2 : JJ = LJK + J
3000 FOR K = JP1 TO NC : KP1 = K + 1
3010 KM1 = K - 1 : JK = LJK + K : KK = KM1 * NC - KM1 * K / 2 + K
3020 EA = (AR(JK) * AR(JK)) / (AR(JJ) * AR(KK))
3030 EB = (BR(JK) * BR(JK)) / (BR(JJ) * BR(KK))
3040 IF EA < EPS AND EB < EPS THEN 3540
3050 '
3060 '  IF ZEROING IS REQUIRED CALCULATE THE ROTATION MATRIX
3070 '  ELEMENT CA AND CG
3080 '
3090 AK = AR(KK) * BR(JK) - BR(KK) * AR(JK)
3100 AJ = AR(JJ) * BR(JK) - BR(JJ) * AR(JK)
3110 AB = AR(JJ) * BR(KK) - BR(JJ) * AR(KK)
3120 CHECK = (AB * AB + 4 * AK * AJ) / 4
3130 IF CHECK < 0 THEN STOP
```

```
3140 SQCH = SQR (CHECK) : D1 = AB / 2 + SQCH : D2 = AB / 2 - SQCH
3150 IF ABS (D2) > ABS (D1) THEN DEN = D2 ELSE DEN = D1
3160 IF DEN = 0 THEN CA = 0 : CG = - AR(JK) / AR(KK) : GOTO 3220
3170 CA = AK / DEN : CG = - AJ / DEN
3180 '
3190 '  PREFORM THE GENERALIZED ROTATION TO ZERO THE PRESENT OFF
3200 '  DIAGONAL ELEMENT
3210 '
3220 IF NC - 2 = 0 THEN 3430
3230 IF JM1 - 1 < 0 THEN 3300
3240 FOR I = 1 TO JM1 : IM1 = I - 1
3250 IJ = IM1 * NC - IM1 * I / 2 + J
3260 IK = IM1 * NC - IM1 * I / 2 + K
3270 AJ = AR(IJ) : BJ = BR(IJ) : AK = AR(IK) : BK = BR(IK)
3280 AR(IJ) = AJ + CG * AK : BR(IJ) = BJ + CG * BK
3290 AR(IK) = AK + CA * AJ : BR(IK) = BK + CA * BJ : NEXT I
3300 IF KP1 - NC > 0 THEN 3360
3310 LJI = JM1 * NC - JM1 * J / 2 : LKI = KM1 * NC - KM1 * K / 2
3320 FOR I = KP1 TO NC : JI = LJI + I : KI = LKI + I
3330 AJ = AR(JI) : BJ = BR(JI) : AK = AR(KI) : BK = BR(KI)
3340 AR(JI) = AJ + CG * AK : BR(JI) = BJ + CG * BK
3350 AR(KI) = AK + CA * AJ : BR(KI) = BK + CA * BJ : NEXT I
3360 IF JP1 - KM1 > 0 THEN 3430
3370 LJI = JM1 * NC - JM1 * J / 2
3380 FOR I = JP1 TO KM1 : JI = LJI + I : IM1 = I - 1
3390 IK = IM1 * NC - IM1 * I / 2 + K
3400 AJ = AR(JI) : BJ = BR(JI) : AK = AR(IK) : BK = BR(IK)
3410 AR(JI) = AJ + CG * AK : BR(JI) = BJ + CG * BK
3420 AR(IK) = AK + CA * AJ : BR(IK) = BK + CA * BJ : NEXT I
3430 AK = AR(KK) : BK = BR(KK)
3440 AR(KK) = AK + 2 * CA * AR(JK) + CA * CA * AR(JJ)
3450 BR(KK) = BK + 2 * CA * BR(JK) + CA * CA * BR(JJ)
3460 AR(JJ) = AR(JJ) + 2 * CG * AR(JK) + CG * CG * AK
3470 BR(JJ) = BR(JJ) + 2 * CG * BR(JK) + CG * CG * BK
3480 AR(JK) = 0 : BR(JK) = 0
3490 '
3500 '  UPDATE THE EIGENVECTOR MATRIX AFTER ROTATION
3510 '
3520 FOR I = 1 TO NC : XJ = VEC(I,J) : XK = VEC(I,K)
3530 VEC(I,J) = XJ + CG * XK : VEC(I,K) = XK + CA * XJ : NEXT I
3540 NEXT K,J
3550 '
3560 '  UPDATE THE EIGENVALUES AFTER SWEEP
3570 '
3580 II = 1
3590 FOR I = 1 TO NC
3600 IF AR(II) < = 0 OR BR(II) < = 0 THEN STOP
3610 EIGV(I) = AR(II) / BR(II) : II = II + N1 - I : NEXT I
3620 IF LP = 0 THEN 3680
3630 PRINT "CURRENT EIGENVALUES IN JACOBI ARE "
3640 FOR I = 1 TO NC : PRINT "EIGV(";I;") = ";EIGV(I) : NEXT I
3650 '
3660 '  CHECK FOR CONVERGENCE
3670 '
```

```
3680 FOR I = 1 TO NC
3690 TL = RT * TT(I) : DI = ABS (EIGV(I) - TT(I))
3700 IF DI > TL THEN 3970
3710 NEXT I
3720 '
3730 '  CHECK ALL OFF-DIAGONAL ELEMENTS TO SET IF ANOTHER
3740 '  SWEEP IS REQUIRED
3750 '
3760 EPS = RT ^ 2
3770 FOR J = 1 TO NR : JM1 = J - 1 : JP1 = J + 1
3780 LJK = JM1 * NC - JM1 * J / 2 : JJ = LJK + J
3790 FOR K = JP1 TO NC : KM1 = K - 1 : JK = LJK + K
3800 KK = KM1 * NC - KM1 * K / 2 + K
3810 EA = (AR(JK) * AR(JK)) / (AR(JJ) * AR(KK))
3820 EB = (BR(JK) * BR(JK)) / (BR(JJ) * BR(KK))
3830 IF EA < ESP AND EB < ESP THEN 3840 ELSE 3970
3840 NEXT K,J
3850 '
3860 '  FILL OUT BOTTOM TRIANGLE OF RESULTANT MATRICES AND
3870 '  SCALE EIGENVECTORS
3880 '
3890 II = 1
3900 FOR I = 1 TO NC : BB = SQR(BR(II))
3910 FOR K = 1 TO NC : VEC(K,I) = VEC(K,I) / BB : NEXT K
3920 II = II + N1 - I : NEXT I
3930 GOTO 4020
3940 '
3950 '  UPDATE TT MATRIX AND START NEW SWEEP, IF ALLOWED
3960 '
3970 FOR I = 1 TO NC : TT(I) = EIGV(I) : NEXT I
3980 IF NSWEEP < NSMAX THEN 2920 ELSE 3890
3990 '
4000 '  END OF JACOBI
4010 '
4020 IF LP = 0 THEN 4070 ELSE IND = 2
4030 PRINT " AR AND BR AFTER JACOBI DIAGONALZATION " : GOTO 2640
4040 '
4050 '  ARRANGE EIGENVALUES IN ASCENDING ORDER
4060 '
4070 IS = 0 : II = 1
4080 FOR I = 1 TO NR : ITEMP = II + N1 - I
4090 IF EIGV(I + 1) > = EIGV(I) THEN 4120
4100 IS = IS + 1 : SWAP EIGV(I),EIGV(I + 1) : SWAP BR(ITEMP),BR(II)
4110 FOR K = 1 TO NC : SWAP VEC(K,I),VEC(K,I + 1) : NEXT K
4120 II = ITEMP : NEXT I
4130 IF IS > 0 THEN 4070
4140 IF LP = 0 THEN 4210
4150 PRINT " EIGENVALUES OF AR- LAMBDA*BR"
4160 FOR I = 1 TO NC : PRINT "EIGV(";I;") = ";EIGV(I) : NEXT I
4170 '
4180 '  CALCULATE B TIMES APPROXIMATE EIGENVECTORS (IV.EQ.0)
4190 '  OR FINAL EIGENVECTOR APPROXIMATES (IV.GT.0)
4200 '
4210 FOR I = 1 TO NN : FOR II = 1 TO NN2
4220 IF I < = NN1 * II THEN IN2 = II : GOTO 4240
```

```
4230 NEXT II
4240 I1 = I - NN1 * (IN2 - 1 ) : FOR J = 1 TO NC
4250 GET #4,(J - 1 ) * NN2 + IN2 : W(J) = CVD(GM$(I1)) : NEXT J
4260 FOR K = 1 TO NC : TR = 0 : GET #4,(K - 1 ) * NN2 + IN2
4270 FOR L = 1 TO NC : TR = TR + W(L) * VEC(L,K) : NEXT L
4280 LSET GM$(I1) = MKD$(TR) : PUT #4,(K - 1) * NN2 + IN2 : NEXT K,I
4290 IF IV > 0 THEN 4480
4300 '
4310 '  CHECK FOR CONVERGENCE OF EIGENVALUES
4320 '
4330 FOR I = 1 TO NC : DI = ABS(EIGV(I) - D(I))
4340 W(I) = DI / EIGV(I) : NEXT I
4350 IF LP = 0 THEN 4390
4360 PRINT "RELATIVE TOLERANCE REACHED ON EIGENVALUES"
4370 FOR I = 1 TO NC : PRINT "W(";I;") = ";W(I) : NEXT I
4380 '
4390 FOR I = 1 TO NC
4400 IF W(I) > RT THEN 4430
4410 NEXT I
4420 PRINT "CONVERGENCE REACHED FOR RTOL = ";RT : IV = 1 : GOTO 1390
4430 IF IT > = NITEM THEN PRINT "NO CONVERGENCE" : IV = 2 : FS = 0 : GOTO 1390
4440 FOR I = 1 TO NC : D(I) = EIGV(I) : NEXT I : GOTO 1390
4450 '
4460 '  ------END OF ITERATION LOOP
4470 '
4480 PRINT : PRINT " THE CALCULATED EIGENVALUES ARE "
4490 PRINT " ============================== "
4500 FOR I = 1 TO NNC : PRINT "EIGV(";I;") = ";EIGV(I) : NEXT I
4510 PRINT : PRINT " THE CALCULATED EIGENVECTORS ARE "
4520 PRINT " ============================== "
4530 FOR J = 1 TO NNC : IN2 = 0 : FOR K1 = 1 TO NN2
4540 GET #4,(J - 1) * NN2 + K1 : FOR K2 = 1 TO NN1
4550 IN2 = IN2 + 1 : IF IN2 > NN THEN 4570
4560 PRINT "R(";IN2;",";J;") = ";CVD(GM$(K2)) : NEXT K2,K1
4570 NEXT J : CLOSE
4580 OPEN "O",#1,"DUMY" : PRINT #1,Q$
4590 PRINT #1,NN;MAXC;NC;NN1;NN2;NH1
4600 FOR I = 1 TO NN + 1 : PRINT #1,MAXA(I) : NEXT I
4610 FOR I = 1 TO NC : PRINT #1,EIGV(I) : NEXT I : CLOSE
4620 CHAIN "MODAL.BAS" : END
```

```
1000 '            ************************
1010 '            ****       MODAL      ****
1020 '            ************************
1030 '
1040 ' ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
1050 ' ::            = SUBPROGRAM  MODAL =                    ::
1060 ' ::  THIS SUBPROGRAM IS DESIGNED TO THE RESPONSE OF     ::
1070 ' ::  A MULTI-DEGREE-OF-FREEDOM SYSTEM BY MODAL          ::
1080 ' ::  SUPERPOSITION METHOD.                              ::
1090 ' ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
1100 '
1110 CLEAR : PRINT CHR$(12) : LOCATE 12,20
1120 PRINT " MODAL SUPERPOSITION MATHOD"
1130 '
1140 DEFDBL A-P,R-Z : DEFINT I-N : DEFSTR Q
1150 '
1160 '  READ DATA
1170 '
1180 DIM MAXA(750),EIGV(31)
1190 OPEN "I",#1,"DUMY" : INPUT #1,Q$
1200 INPUT #1,NN,MAXC,NC,NN1,NN2,NH1
1210 FOR I = 1 TO NN + 1 : INPUT #1,MAXA(I) : NEXT I
1220 FOR I = 1 TO NC : INPUT #1,EIGV(I) : NEXT I : CLOSE
1230 '
1240 '  STATEMENT FUNCTION
1250 '
1260 DEF FNT1(T) = EXP(XIWD * T)*(XIWD * COS(WD * T)+ WD * SIN(WD * T))/DWSQ
1270 DEF FNT2(T) = EXP(XIWD * T)*(XIWD * SIN(WD * T)- WD * COS(WD * T))/DWSQ
1280 DEF FNT3(T) = T * FNT2(T) - XIWD * FNT2(T) / DWSQ + WD * FNT1(T) / DWSQ
1290 DEF FNT4(T) = T * FNT1(T) - XIWD * FNT1(T) / DWSQ - WD * FNT2(T) / DWSQ
1300 '
1310 '  READ FORCING FUNCTION NAD INTERPOLATE
1320 '       |
1330 OPEN "I",#1,Q$ : INPUT #1,XI,DT,TMAX
1340 DIM P(4,100),T(4,100),F(750)
1350 DIM NT(4),FM(31,4),II(4),ANN(4)
1360 DIM GG$(31),D$(31)
1370 FOR J = 1 TO NH1 : INPUT #1,NT(J) : FOR I = 1 TO NT(J)
1380 INPUT #1,T(J,I),P(J,I) : NEXT I,J : CLOSE
1390 OPEN "R",#1,"EIGVEC",NN1 * 8 : FOR I = 1 TO NN1
1400 FIELD #1,(I - 1) * 8 AS G$,8 AS GG$(I) : NEXT I
1410 OPEN "O",#2,"FORCE"
1420 NT = INT(TMAX / DT) : NT2 = NT + 2 : NT3 = NT + 3
1440 OPEN "R",#3,"MODISP",NC * 8 : FOR I = 1 TO NC
1450 FIELD #3,(I - 1) * 8 AS G$,8 AS D$(I) : NEXT I
1460 FOR I = 1 TO NC : LSET D$(I) = MKD$(0) : NEXT I
1470 FOR I = 1 TO NT3 : PUT #3,I : NEXT I
1480 OPEN "I",#4,"STFORCE"
1490 FOR I = 1 TO NH1 : PRINT #2,P(I,1)
1500 II(I) = 1 : ANN(I) = 0 : NEXT I
1510 FOR I = 2 TO NT3 : AI = I - 1 : TA = AI * DT
1520 FOR J = 1 TO NH1 : IF II(J) + 1 > NT(J) THEN PRINT #2,0 : GOTO 1590
1530 IF TA < = T(J,II(J) + 1) THEN 1560
1540 ANN(J) = - T(J,II(J) + 1) + TA - DT : II(J) = II(J) + 1
1550 IF II(J) + 1 > NT(J) THEN PRINT #2,0 : GOTO 1590
```

```
1560 ANN(J) = ANN(J) + DT : TT = T(J,II(J) + 1) - T(J,II(J))
1570 FF = P(J,II(J)) + (P(J,II(J) + 1) - P(J,II(J))) * ANN(J) / TT
1580 PRINT #2,FF
1590 NEXT J,I : CLOSE #2
1600 '
1610 '  DETERMINE TIME AND EQUIVALENT FORCES
1620 '
1630 FOR K = 1 TO NH1 : FOR J = 1 TO NN : INPUT #4,F(J) : NEXT J
1640 FOR ID = 1 TO NC : IN = 0 : FOR I = 1 TO NN2
1650 GET #1,(ID - 1) * NN2 + I : FOR J = 1 TO NN1
1660 IN = IN + 1 : FM(ID,K) = FM(ID,K) + F(IN) * CVD(GG$(J))
1670 IF IN = NN THEN 1690
1680 NEXT J,I
1690 NEXT ID,K : CLOSE #4 : FOR ID = 1 TO NC
1700 OPEN "I",#2,"FORCE" : FIM1 = 0
1710 FOR J = 1 TO NH1 : INPUT #2,F1
1720 FIM1 = FIM1 + F1 * FM(ID,J) : NEXT J
1730 '
1740 '  SET INITIAL VALUES
1750 '
1760 TIM1 = 0 : ATI = 0 : BTI = 0 : AM = 1 : AK = EIGV(ID)
1770 OMEGA = SQR (AK / AM) : CRIT = 2 * SQR (AK * AM)
1780 C = XI * CRIT : WD = OMEGA * SQR (1 - (XI * XI))
1790 XIWD = XI * OMEGA : DWSQ = XIWD * XIWD + WD * WD
1800 '
1810 ' LOOP OVER TIME AND SOLVE FOR MODAL DISPLACEMENT
1820 '
1830 FOR I = 1 TO NT2 : GET #3,I
1840 FI = 0 : FOR J = 1 TO NH1 : INPUT #2,F1
1850 FI = FI + F1 * FM(ID,J) : NEXT J
1860 TI = DT * I : DFTI = FI - FIM1
1870 DTI = TI - TIM1 : FT = DFTI / DTI : G = FIM1 - TIM1 * FT
1880 AI = FNT1(TI) - FNT1(TIM1) : BI = FNT2(TI) - FNT2(TIM1)
1890 VS = FNT3(TI) - FNT3(TIM1) : VC = FNT4(TI) - FNT4(TIM1)
1900 AI = AI * G : AI = AI + FT * VC : ATI = ATI + AI
1910 BI = BI * G : BI = BI + FT * VS : BTI = BTI + BI
1920 YY = ATI * SIN(WD * TI) - BTI * COS(WD * TI)
1930 Y = EXP( - XIWD * TI) * YY / (AM * WD)
1940 TIM1 = TI : FIM1 = FI : LSET D$(ID) = MKD$(CVD(D$(ID)) + Y)
1950 PUT #3,I : NEXT I : CLOSE #2 : NEXT ID : CLOSE
1960 '
1970 OPEN "O",#1,"DUMY"
1980 PRINT #1,NN;NC;MAXC;NH1;NT;DT;NN1;NN2;XI
1990 FOR I = 1 TO NN + 1 : PRINT #1,MAXA(I) : NEXT I : CLOSE
2000 CHAIN "D-OUT.BAS" : END
```

```
1000 '            **************************
1010 '            ****       D-OUT      ****
1020 '            **************************
1030 '
1040 ' ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
1050 ' ::            = SUBPROGRAM  D-OUT =                    ::
1060 ' ::  THIS SUBPROGRAM COMPUTES RESPONSE OF DYNAMIC       ::
1070 ' ::  ANALYSIS AND PRINT OUT AT EACH TIME STEP.          ::
1080 ' ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
1090 '
1100 CLEAR : PRINT CHR$(12) : LOCATE 12,20
1110 PRINT " PRINT OUT DATA "
1120 '
1130 DEFDBL A-P,R-Z : DEFINT I-N : DEFSTR Q
1140 '
1150 '  READ DATA
1160 '
1170 DIM MAXA(750) : OPEN "I",#1,"DUMY"
1180 INPUT #1,NN,NC,MAXC,NH1,NT,DT,NN1,NN2,XI
1190 FOR I = 1 TO NN + 1 : INPUT #1,MAXA(I) : NEXT I : CLOSE
1200 '
1210 OPEN "I",#1,"NDOF" : INPUT #1,NM,NJ
1220 DIM NDOF(250,3),GG$(31),D$(31)
1230 DIM UD(600,2),Y(31),DI(600,5)
1240 DIM ANN(4),AA(27),D(6),F(6),LF(6)
1250 DIM GS$(21),FF$(30)
1260 FOR I = 1 TO NJ : FOR J = 1 TO 3
1270 INPUT #1,NDOF (I,J) : NEXT J,I : CLOSE
1280 OPEN "R",#1,"EIGVEC",NN1 * 8 : FOR I = 1 TO NN1
1290 FIELD #1,(I - 1) * 8 AS G$,8 AS GG$(I) : NEXT I
1300 OPEN "I",#2,"FORCE" : FOR I = 1 TO NH1 : INPUT #2,FI : NEXT I
1310 OPEN "R",#3,"MODISP",NC * 8 : FOR I = 1 TO NC
1320 FIELD #3,(I - 1) * 8 AS G$,8 AS D$(I) : NEXT I
1330 FOR I = 1 TO 5 : FOR J = 1 TO NN : DI(J,I) = 0 : NEXT J,I
1340 B$ =  "###      X      #######.####  #######.####  #######.####"
1350 C$ =  "         Y      #######.####  #######.####  #######.####"
1360 D$ =  "      ROTATn.   #######.####  #######.####  #######.####"
1370 E$ =  " ####   ###     #######.####  #######.####  #######.####"
1380 F$ =  "        ###     #######.####  #######.####  #######.####"
1390 '
1400 '  CALCULATE AND PRINT OUT NODAL RESPONSE
1410 '
1420 NT2 = NT + 2 : FOR IT = 3 TO 4 : GET #3,IT - 2
1430 FOR I = 1 TO NC : Y(I) = CVD(D$(I)) : NEXT I
1440 FOR I = 1 TO NN : FOR II = 1 TO NN2
1450 IF II * NN1 > = I THEN IN = II : GOTO 1470
1460 NEXT II
1470 FOR J = 1 TO NC : GET #1,(J - 1) * NN2 + IN
1480 I1 = I - NN1 * (IN - 1) : X = CVD(GG$(I1))
1490 DI(I,IT) = DI(I,IT) + X * Y(J) : NEXT J,I,IT
1500 FOR I = 1 TO NN : DI(I,1) = DI(I,3) :NEXT I
1510 FOR IT = 1 TO NT
1520 GET #3,IT + 2
1530 FOR I = 1 TO NC : Y(I) = CVD(D$(I)) : NEXT I
1540 FOR I = 1 TO NH1 : INPUT #2,ANN(I) : NEXT I
1550 PRINT : PRINT "  TIME = ";IT * DT
```

```
1560 PRINT " ==============="
1570 A$ =  "*********************************************************"
1580 PRINT : PRINT A$
1590 PRINT "***       NODAL RESPONSE IN GLOBAL COORDINATES        ***"
1600 PRINT A$ : PRINT
1610 PRINT "NODE  DIRCETn.  DISPLACEMENT      VELOCITY  ACCERELATION"
1620 PRINT "-----------------------------------------------------"
1630 FOR I = 1 TO NN : DI(I,5) = 0 : FOR II = 1 TO NN2
1640 IF II * NN1 > = I THEN IN = II : GOTO 1660
1650 NEXT II
1660 FOR J = 1 TO NC : GET #1,(J - 1) * NN2 + IN
1670 I1 = I - NN1 * (IN - 1) : X = CVD(GG$(I1))
1680 DI(I,5) = DI(I,5) + X * Y(J) : NEXT J,I
1690 '
1700 '  CENTRAL FINITE DIFFERENCE
1710 '
1720 FOR J = 1 TO NN
1730 UD(J,1) = (DI(J,1) - 8 * (DI(J,2) - DI(J,4)) - DI(J,5)) / (12 * DT)
1740 AC = -1 * DI(J,1) + 16 * (DI(J,2) + DI(J,4)) - 30 * DI(J,3) - DI(J,5)
1750 UD(J,2) = AC / (12 * DT * DT)
1760 NEXT J
1770 FOR I = 1 TO 4 : FOR J = 1 TO NN : II = I + 1
1780 DI(J,I) = DI(J,II) : NEXT J,I : GOTO 1830
1790 FOR J = 1 TO NN
1800 UD(J,1) = (-3 * DI(J,2) + 4 * DI(J,3) - DI(J,4)) / (2 * DT)
1810 AC = 2 * DI(J,2) - 5 * DI(J,3) + 4 * DI(J,4) - DI(J,5)
1820 UD(J,2) = AC / (DT * DT) : NEXT J
1830 FOR I = 1 TO NJ
1840 IF NDOF(I,1) = 0 THEN PRINT USING B$;I;0;0;0 : GOTO 1860
1850 K = NDOF(I,1) : PRINT USING B$;I;DI(K,2);UD(K,1);UD(K,2)
1860 IF NDOF(I,2) = 0 THEN PRINT USING C$;0;0;0 : GOTO 1880
1870 K = NDOF(I,2) : PRINT USING C$;DI(K,2);UD(K,1);UD(K,2)
1880 IF NDOF(I,3) = 0 THEN PRINT USING D$;0;0;0 : GOTO 1900
1890 K = NDOF(I,3) : PRINT USING D$;DI(K,2);UD(K,1);UD(K,2)
1900 NEXT I
1910 '
1920 '  CALCULATE AND PRINT OUT MEMBER FORCE
1930 '
1940 PRINT : PRINT A$
1950 PRINT "***       MEMBER FORCES IN LOCAL COORDINATES         ***"
1960 PRINT A$ : PRINT
1970 PRINT "MEMBER  NODE     AXIAL FORCE   SHEAR FORCE       MOMENT"
1980 PRINT "-----------------------------------------------------"
1990 OPEN "R",#4,"LOSTIFF",168 : OPEN "R",#5,"LOFORCE",NH1 * 48 + 20
2000 FOR I = 1 TO 21
2010 FIELD #4,(I - 1) * 8 AS A$,8 AS GS$(I) : NEXT I
2020 FIELD #5,8 AS D1$,8 AS D2$,2 AS DJ$,2 AS DK$
2030 FOR I = 1 TO NH1 * 6
2040 FIELD #5,(I - 1) * 8 + 20 AS A$,8 AS FF$(I) : NEXT I
2050 FOR LZ = 1 TO NM : GET #4,LZ : GET #5,LZ
2060 FOR I = 1 TO 27 : AA(I) = 0 : NEXT I
2070 X1 = CVD(D2$) : X2 = CVD(D1$) : X3 = 0 : Y1 = - X2
2080 Y2 = X1 : Y3 = 0 : Z1 = 0 : Z2 = 0 : Z3 = 1
2090 JQ = CVI(DJ$) : KQ = CVI(DK$)
```

```
2100 FOR I = 1 TO 21 : AA(I) = CVD(GS$(I)) : NEXT I
2110 FOR I = 1 TO 6 : FOR J = 1 TO NH1 : JJ = (J - 1) * 6 + I
2120 AA(21 + I) = AA(21 + I) + ANN(J) * CVD(FF$(JJ)) : NEXT J,I
2130 FOR I = 1 TO 3 : IF NDOF(JQ,I) = 0 THEN D(I) = 0 : GOTO 2150
2140 D(I) = DI(NDOF(JQ,I),2)
2150 IF NDOF(KQ,I) = 0 THEN D(I + 3) = 0 : GOTO 2170
2160 D(I + 3) = DI(NDOF(KQ,I),2)
2170 NEXT I
2180 FOR I = 1 TO 6 STEP 3 : R1 = D(I)
2190 R2 = D(I + 1) : R3 = D(I + 2)
2200 D(I) = R1 * X1 + R2 * X2 + R3 * X3
2210 D(I + 1) = R1 * Y1 + R2 * Y2 + R3 * Y3
2220 D(I + 2) = R1 * Z1 + R2 * Z2 + R3 * Z3
2230 NEXT I
2240 ' CALCULATE MEMBER FORCES IN LOCAL COORDINATES
2250 LF(1) = 1 : LF(2) = 7 : LF(3) = 12
2260 LF(4) = 16 : LF(5) = 19 : LF(6) = 21
2270 FOR I = 1 TO 6 : F(I) = 0 : IO = LF(I) - I
2280 FOR J = 1 TO 6 : JO = LF(J) - J
2290 IF LF(J) > = LF(I) THEN 2310
2300 F(I) = F(I) + AA(JO + I) * D(J) : GOTO 2320
2310 F(I) = F(I) + AA(IO + J) * D(J)
2320 NEXT J : F(I) = F(I) - AA(21 + I) : NEXT I
2330 PRINT USING E$;LZ;JQ;F(1);F(2);F(3)
2340 PRINT USING F$;KQ;F(4);F(5);F(6)
2350 NEXT LZ : CLOSE #4 : CLOSE #5
2360 NEXT IT
2370 CLOSE : CHAIN "OPTION.BAS" : END
```

```
1000 '            ***************************
1010 '            ****     S-OUT         ****
1020 '            ***************************
1030 '
1040 ' ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
1050 ' ::            = SUBPROGRAM  S-OUT =                    ::
1060 ' ::  THIS SUBPROGRAM COMPUTES RESPONSE OF STATIC        ::
1070 ' ::  ANALYSIS AND PRINT.                                ::
1080 ' ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
1090 '
1100 CLEAR : PRINT CHR$(12) : LOCATE 12,20
1110 PRINT : PRINT " PRINT OUT DATA "
1120 '
1130 DEFDBL A-P,R-Z : DEFINT I-N : DEFSTR Q
1140 '
1150 '  READ DATA
1160 '
1170 DIM NDOF(250,3),FF(750,1) : OPEN "I",#1,"DUMY"
1180 INPUT #1,NN,NJ,NM,NC,NH1
1190 FOR I = 1 TO NJ : FOR J = 1 TO 3
1200 INPUT #1,NDOF(I,J) : NEXT J,I
1210 FOR I = 1 TO NN : INPUT #1,FF(I,1) : NEXT I : CLOSE
1220 '
1230 DIM AA(27),D(6),F(6),LF(6),GS$(21),FF$(30)
1240 OPEN "R",#4,"LOSTIFF",168 : OPEN "R",#5,"LOFORCE",NH1 * 48 + 20
1250 FOR I = 1 TO 21
1260 FIELD #4,(I - 1) * 8 AS A$,8 AS GS$(I) : NEXT I
1270 FIELD #5,8 AS D1$,8 AS D2$,2 AS DJ$,2 AS DK$
1280 FOR I = 1 TO NH1 * 6
1290 FIELD #5,(I - 1) * 8 + 20 AS A$,8 AS FF$(I) : NEXT I
1300 B$ = "###   ########.####   #######.####   #######.####"
1310 A$ = "************************************************"
1320 PRINT : PRINT A$
1330 PRINT "*** NODAL DISPLACEMENTS IN GLOBAL COORDINATES ***
1340 PRINT A$ : PRINT
1350 PRINT "NODE       X-DISPL.       Y-DISPL.       ROTATION"
1360 PRINT "------------------------------------------------"
1370 FOR I = 1 TO NJ
1380 IF NDOF(I,1) = 0 THEN A1 = 0 ELSE A1 = FF(NDOF(I,1),1)
1390 IF NDOF(I,2) = 0 THEN A2 = 0 ELSE A2 = FF(NDOF(I,2),1)
1400 IF NDOF(I,3) = 0 THEN A3 = 0 ELSE A3 = FF(NDOF(I,3),1)
1410 PRINT USING B$;I;A1;A2;A3 : NEXT I
1420 ' COORDINATES INTO LOCAL COORDINATES
1430 C$ =  " ####     ###  #####.####  #####.####  #####.####"
1440 D$ =  "          ###  #####.####  #####.####  #####.####"
1450 PRINT : PRINT A$
1460 PRINT "***    MEMBER FORCES IN LOCAL COORDINATES     ***"
1470 PRINT A$ : PRINT
1480 PRINT "MEMBER    NODE      AXIAL       SHEAR      MOMENT"
1490 PRINT "------------------------------------------------"
1500 FOR LZ = 1 TO NM : GET #4,LZ : GET #5,LZ
1510 FOR I = 1 TO 27 : AA(I) = 0 : NEXT I
1520 X1 = CVD(D2$) : X2 = CVD(D1$) : X3 = 0 : Y1 = - X2
1530 Y2 = X1 : Y3 = 0 : Z1 = 0 : Z2 = 0 : Z3 = 1
1540 JQ = CVI(DJ$) : KQ = CVI(DK$)
```

```
1550 FOR I = 1 TO 21 : AA(I) = CVD(GS$(I)) : NEXT I
1560 FOR I = 1 TO 6 : FOR J = 1 TO NH1 : JJ = (J - 1) * 6 + I
1570 AA(21 + I) = AA(21 + I) + CVD(FF$(JJ)) : NEXT J,I
1580 FOR I = 1 TO 3 : IF NDOF(JQ,I) = 0 THEN D(I) = 0 : GOTO 1600
1590 D(I) = FF(NDOF(JQ,I),1)
1600 IF NDOF(KQ,I) = 0 THEN D(I + 3) = 0 : GOTO 1620
1610 D(I + 3) = FF(NDOF(KQ,I),1)
1620 NEXT I
1630 FOR I = 1 TO 6 STEP 3 : R1 = D(I)
1640 R2 = D(I + 1) : R3 = D(I + 2)
1650 D(I) = R1 * X1 + R2 * X2 + R3 * X3
1660 D(I + 1) = R1 * Y1 + R2 * Y2 + R3 * Y3
1670 D(I + 2) = R1 * Z1 + R2 * Z2 + R3 * Z3
1680 NEXT I
1690 ' CALCULATE MEMBER FORCES IN LOCAL COORDINATES
1700 LF(1) = 1 : LF(2) = 7 : LF(3) = 12
1710 LF(4) = 16 : LF(5) = 19 : LF(6) = 21
1720 FOR I = 1 TO 6 : F(I) = 0 : IO = LF(I) - I
1730 FOR J = 1 TO 6 : JO = LF(J) - J
1740 IF LF(J) > = LF(I) THEN 1760
1750 F(I) = F(I) + AA(JO + I) * D(J) : GOTO 1770
1760 F(I) = F(I) + AA(IO + J) * D(J)
1770 NEXT J : F(I) = F(I) - AA(21 + I) : NEXT I
1780 J = JQ : PRINT USING C$;LZ;J;F(1);F(2);F(3)
1790 K = KQ : PRINT USING D$;K;F(4);F(5);F(6)
1800 NEXT LZ
1810 CLOSE : CHAIN "OPTION.BAS" : END
```

ประวัติ

นายวิศิษย์ เพชรภูวดี เกิดเมื่อวันที่ 20 สิงหาคม พ.ศ. 2505 เกิดที่จังหวัดเพชรบุรี สำเร็จการศึกษามัธยมศึกษาตอนปลาย จากโรงเรียนอรุณประดิษฐ์ ในเดือนมีนาคม พ.ศ. 2522 สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมโยธา จากมหาวิทยาลัยขอนแก่น ในเดือนมีนาคม พ.ศ. 2526 ขณะนี้ทำงานอยู่ที่ กรมทางหลวงแห่งประเทศไทย