"its use in a typical application in the field of structural analysis is illustrated.

An Interactive Matrix Arithmetic Module (IMAM)

by

Salvador F. Reyes*

Introduction

This paper describes an attempt to exploit the usefulness of the microcomputer as a learning tool through the development of an applications program module which carries out various common operations on vectors and matrices (with real elements) in an interactive fashion and without unduly burdening the user with "housekeeping" details. Although the selection of matrix operations in the module gears it for applications in linear structural analysis, the package may be a useful learning aid for introductory courses on matrix analysis as well.

To provide a high degree of portability the source code for the Interpretive Matrix Arithmetic Module (IMAM) described is written in Microsoft Interpreter BASIC. A display unit and a disk drive are required peripherals. A line printer is optional.

IMAM features menu drivers which enable the user to direct the program to undertake the desired sequence of operations or tasks through menu selections. In the course of the execution of certain tasks the user is prompted for appropriate responses such as data entries, option selections, etc., and provisions are made for corrective responses or error exits where required or convenient. Another major feature of IMAM is that it can be extended by appending modules which may themselves make use of IMAM facilities. Conventions to be observed in generating such extensions of the program are described herein.

Program Description and Usage

IMAM is essentially a facility for setting up and performing arithmetic operations on a set of nine matrices. Labelled A, B, . . ., H, and S, their sizes (number of rows/columns) and element values are input or modified as required in the initiation or execution of tasks.

Arithmetic operations are stylized (as displayed in the menus) and generally involve matrices B and C as operands. The result of the operation is received by matrix A in all cases. Some unary operations involve B and send the result to A or are carried out on A itself. Special matrix operations may also involve S (i.e., modify it). All other matrices are defined or altered only by specific assignments. (However, matrix B is altered in the solution of the eigenvalue problem – see below). Matrices D, . . ., H, are available to hold intermediate results arising from the evaluation of a complex arithmetic expression. Data transfers between these matrices and A (as well as between A and S) are effected by menu selections.

*Professor of Civil Engineering, University of the Philippines.

The main facilities of IMAM are listed in a main or primary menu which is displayed at the outset after each completion of a task or computation. Of the ten items in the main menu, five actually represent task groups for which the respective (secondary) menus are displayed on selection. The items in the main menu are the following:

00/0 –	Line Print a Message
01/0 –	Input/Output/File Management
02/0 –	Assignment (B, . . ., H, S to/from A)
03/0 –	Arithmetic Operation
04/0 –	Special Matrix Operation
05/0 –	Perform a Utility Function
06/0 –	Create a User Module
07/0 –	Enter a User Module
08/0 –	End Problem Run
09/0 –	End Session

When any one of the five items, 01/0 to 05/0, is selected from the main menu the corresponding secondary menu is displayed and any task selected from that menu is then immediately carried out. If task 06/0 is selected, control passes to the BASIC interpreter. Task 07/0 passes control to the user module.

The specific tasks included in each of the five secondary menus are listed below. Most are self-explanatory and the user is prompted for any action that may be required in connection with the execution of the current task.

Input/Output/File Management

The tasks included in this group are

01/1 –	Input Elements of A (by Rows)
02/1 –	Input Selected Elements of A
03/1 –	Display A
04/1 –	Display S
05/1 –	Print A
06/1 –	Print S
07/1 –	ERASE Indicated Matrices (A, . . ., H,S)
08/1 –	A or S → FILE
09/1 –	A or S ← FILE

Basically, sizes of matrices and initializations can be done for matrices A and S only. Only matrix A accepts element inputs from the keyboard. Hence, other matrices will have to be set as required through assignment statements (see below).

The backing store or file is created or updated on the user's option. The assumed filename is "IMAMIDAT.DAT". Whenever a matrix is saved or filed, an identification field must be given. Any substring of alpha-numeric characters from the leading portion of this field may be used as a key field for subsequent retrievals of the filed matrix. The maximum permissible length of the identification field (including blanks) is 72. Matrices may be deleted from the file and the resulting "holes" may be reallocated for new matrices. A "garbage collection" or storage compaction function is available.

Assignments

Assignment statements instruct IMAM to transfer data to or from A and the other matrices (or to initialize A or S).

01/2 –	[0] → A	11/2 –	S ← [0]
02/2 –	A → B	12/2 –	A ← B
03/2 –	A → C	13/2 –	A ← C
04/2 –	A → D	14/2 –	A ← D
05/2 –	A → E	15/2 –	A ← E
06/2 –	A → F	16/2 –	A ← F
07/2 –	A → G	17/2 –	A ← G
08/2 –	A → H	18/2 –	A ← H
09/2 –	A → S	19/2 –	A ← S
10/2 –	A → S[p,q]	20/2 –	A ← S [p,q]

Statements 10/2 and 20/2 are to be used for assigning or retrieving submatrices of S. In these statements the indices, p and q, are those of the pivot element corresponding to A(1,1). The size of the submatrix is that of A. When these statements are selected the user will be allowed to define sizes and initialize the matrices as desired.

Arithmetic Operations

The following statements comprise the set of standard arithmetic operations:

01/3 –	A ← B * C
02/3 –	A ← B + C
03/3 –	A ← B – C
04/3 –	A ← Transpose of B
05/3 –	A ← c * A (c a scalar)
06/3 –	A ← Negative of A
07/3 –	A ← Inverse of A
08/3 –	Solve S X = A for X (result to A)
09/3 –	Eigen VP ([A] – e [B]); e's → Diag [A], Vects → Cols [S]

In all cases the result is installed in matrix A and the values of the other matrices involved, if any, are unaltered. However, in the solution of simultaneous equations the coefficient matrix (S) is erased. The eigenvalue problem solver alters matrix B.

Special Matrix Operations

Included among the following special matrix operations are some that commonly occur in structural analysis.

01/4 –	'TO' Row Permutation, A (I,.) → A (P (I), .)
02/4 –	'FROM' Row Permutation, A (P (I), .) → A (I,.)
03/4 –	'TO' Column Permutation, A (., I) → A (., P (I))
04/4 –	'FROM' Column Permutation, A (., P (I)) → A (., I)
05/4 –	A = B' * C * B
06/4 –	S [p, q] = S [p,q] + B' * C * B
07/4 –	S [p, q] = S [p,q] + A [r, s; nr, ns]
08/4 –	S [p, q] = A [r, s; nr, ns]

The last two operations provide means for transferring and incrementing elements on a submatrix to submatrix basis: p, q and r, s are pivotal indices and nr, ns are, respectively, the number of rows and number of columns of the submatrices involved.

Utilities

The following self-explanatory "housekeeping" tasks are currently implemented:

01/5 – DISPLAY Sizes of Matrices and Free Area
02/5 – DISPLAY/LIST Commands Executed in Current Run
03/5 – DISPLAY/LIST Directory of Filed Matrices
04/5 – COMPACT File of Matrices (Garbage Collection)

IMAM Extensions

The basic module can be augmented by user-generated modules subject to the following restrictions:

1. To avoid conflicts with variable name usage in the main module the user module must use only two-character "local" variable names and the second character must be a numeral. IMAM issues a DEFINT I–N statement on initialization.

2. The entry point in the user module must be a source BASIC statement numbered 10000. Return of control to IMAM must be a GO TO 210 statement.

If IMAM facilities are to be used by the user module (as described below) the following additional restrictions must be observed:

1. Items in the primary menu are not to be called from a user module.
2. 'FROM' permutations are not operable from a user module.

The user module may assign the following functions to IMAM by using appropriate GOSUB [statement number] statements.

(a) Allocating storage for system matrices and assigning values to their elements.
(b) Performing arithmetic and other operations on the matrices.
(c) Setting displays and halting processing.

3. To suppress messages normally issued by the main module in the course of the execution of the various tasks, set PQ % = 2.

Allocating and Setting Values Matrices

To allocate storage space for a matrix the required statement is

GOSUB 4710.

to be issued after setting parameter values as follows:

MM – an integer between 0 and 8 to tag the matrix for which space is to be allocated (0 tags matrix A, 1 matrix B, etc.)
NR – number of rows
NC – number of columns

Allocating space for a matrix automatically destroys its previous value (if any) and sets the elements of the new matrix definition to zeroes. To de-allocate (erase) a matrix, issue

```
GOSUB 4590
```

after setting the parameters as in the operation of allocation.

Matrix Storage Modes

All system matrices except A and S are stored in the normal BASIC mode with beginning row/column indices of 0 and highest valid indices of NR−1 and NC−1. The first row/column of matrices A and S are used as work vectors by IMAM, which allocates the required extra row and column. The beginning indices are therefore 1 and the highest valid indices NR and NC.

It is sometimes necessary to retrieve the dimensions of a system matrix. For this purpose the statement

```
GOSUB 4890
```

returns the number of rows in NR and the number of columns in NC. If the tagged matrix did not exist IMAM assigns EE%=2 signifying that the matrix was null.

Arithmetic and Other Operations

Except for the group items and operations in the primary menu (which are meaningless at user level) and permutation commands 2/4 and 4/4, all IMAM commands can be accessed with a

```
GOSUB 640
```

statement after setting parameter values. The universally required parameters are

PP% – Task Group Number
KK – Task Number

The operations listed above indicate the proper values in all cases; e.g., for matrix multiplication a possible sequence of BASIC statements is

```
PP% = 3:KK = 1:GOSUB 640
```

For certain operations other parameters have to be set. The tasks involving submatrices require the pivot element indices p and q. Tasks 07/4 and 08/4 also require the pivot indices r and s of the A matrix, and the size (nr, ns) of the submatrix addressed. IMAM expects in these cases that the values are contained in the predefined array IX in the order mentioned; i.e., IX (1) must contain p, IX (2), q, etc. IX (0) is not utilized.

In scalar mutliplications IMAM expects the scalar multiplier to be stored in A (0, 0). In executing 'TO' permutation commands IMAM expects to find the pointers in vector A (., 0) for row permutations and in vector A (0, .) for column permutations. No validity checks are made. Other parameters will have to be keyed in by the user in response to the usual IMAM prompts.

Displays and Prompts

An IMAM prompter displays a message and waits for the user's response. Entry points, displayed messages, and functions (actions based on the keyed in responses) are as follows

Entry Point	Message	Action

5830	*Press [1] if YES, [0] if NO	Sets
		JJ=0 if no;
		JJ=1 if yes

7340	*Invalid Input. RE-TRY? *Press [1] if YES, [0] if NO	–ditto–
7420	*CONTINUE? *Press [1] if YES, [0] if NO	–ditto–

The following two examples illustrate most of the features discussed. The first is solved at IMAM level while a user module is developed for the second.

Example

The eigenvalue problem

$$A\,X = e\,B\,X$$

in which A and B are symmetric and positive definite arises in stability and dynamic analysis of structures. If the equation is recast as

$$B^{-1}\,A\,X = e\,X$$

the well known method of Stodola (Crandall, 1956) calculates the largest eigenvalue, e, and corresponding eigenvector, X, in an iterative fashion. While the successive matrix-vector multiplications and factoring are operations which can be done manually under IMAM it is convenient to generate a user module. Appendix "A" illustrates a possible implementation where the convenience afforded by using IMAM facilities is apparent.

Conclusion

A matrix interpretive program which can be implemented on most microcomputers has been described and its use in a typical application in the field of structural analysis illustrated. A complete listing of the source program is presented in Appendix "B".

Although the program is primarily intended as a teaching aid it can certainly be used for the occasional solution of practical problems, provided enough RAM is available in the computer. Excluding storage for arrays the program is approximately 25–k bytes in length. In such applications the generation of a supplementary or user module similar to that illustrated may be helpful.

Reference

Crandall, S. H. (1956). ENGINEERING ANALYSIS, A Survey of Numerical Procedures, McGraw-Hill, N.Y.

Appendix A
TYPICAL IMAM USER MODULE

```
10000 PRINT CHR$(140)
10010 PRINT "*Module Name: IUMSTODV/BAS":PRINT
10020 PRINT "*Usage: Calculation of the Largest Eigenvalue and Associated Vector
"
10030 PRINT " of the System   HX = ex. :PRINT
10040 PRINT "*Method: Stodola/Vianello Algorithm (Ref: Crandall, 1956)":PRINT
10050 PRINT "*Module Expects Matrix H in [ B ]. Vector [ C ] is used as a Work A
rea.":PRINT
10060 PRINT "*Largest Eigenvalue Returned in A(0) and the Elements of the"
10070 PRINT " Associated Eigenvector in A(1) ... A(N).":PRINT
10080 GOSUB 7420
10090 IF JJ=0 THEN 210
10100 PQ%=2:MM=0:GOSUB 4890
10110 IF EE%>0 THEN 10350
10120 IF NR=NC THEN 10130ELSE EE%=6:GOTO 10350
10130 N1=NR:NC=1:MM=2:GOSUB 4720
10140 IF EE%>0 THEN 10350
10150 FOR I1=0 TO NR-1:C(I1,0)=1:NEXT I1:X1=0:PRINT CHR$(140):J1=0
10160 PP%=3:KK=1:GOSUB 640
10170 T1=0
10180 FOR I1=1 TO N1:IF A(I1,1)>T1 THEN T1=A(I1,1)
10190 NEXT I1
10200 J1=J1+1:A(0,0)=T1:PRINT "*Iteration No. ";J1
10210 IF ABS(T1-X1)/T1<.00001 THEN 10250
10220 A(0,0)=1/T1:PP%=3:KK=5:GOSUB 640
10230 X1=T1:PP%=2:KK=3:GOSUB 640
10240 GOTO 10160
10250 PRINT CHR$(140):PRINT "*Eigenvalue : ";T1:PRINT:PRINT "*Eigenvector:":PRIN
T
10260 X1=0
10270 FOR I1=1 TO N1:X1=X1+A(I1,1)^2:NEXT I1
10280 A(0,0)=1/SQR(X1):PP%=3:KK=5:GOSUB 640
10290 A(0,0)=T1
10300 FOR I1=1 TO N1:PRINT A(I1,1),:NEXT I1:PRINT
10310 PRINT:PRINT "*Print RESULTS?":GOSUB 5830
10320 IF JJ=0 THEN 210
10330 LPRINT:LPRINT "*Eigenvalue : ";T1:LPRINT:LPRINT "*Eigenvector:"
10340 FOR I1=1 TO N1:LPRINT A(I1,1),:NEXT I1:LPRINT:GOTO 210
10350 PQ%=0:GOTO 680
```

Appendix B
IMAM SOURCE LISTING

```
10 'Filename: "IMAM2.BAS"   (CP/M 2.2 Version)
20 '(I)nterpretive (M)atrix (A)rithmetic (M)odules"
30 'Programmed by SFR                    14 July 1984"
40 GOSUB 8480:PRINT "* I    M    A    M *"
50 PRINT "Version 2   sfr 1984"
60 PRINT "All Rights  Reserved":PRINT
70 CLEAR 480:DIM TT%(1,8),IX(6):DEFINT I-N:PRINT"*Specify OPTIONS:":PRINT
80 AA%=0:ON ERROR GOTO 190
90 FK%=0:PRINT"*USE  Backing  Store (FILE)";:GOSUB 5820
100 IF JJ=0 THEN 140
110 OPEN "R",1,"IMAM1DAT.DAT",80:FIELD 1, 80 AS BF$:PRINT "*NEW File";:GOSUB 582
0
120 IF JJ=1 THEN 130 ELSE GET 1,1:FK%=CVI(MID$(BF$,5,2))-1
130 FK%=FK%+1
140 PRINT"*ECHO PRINT issued COMMANDS";:GOSUB 5820
150 ZP%=0:IF JJ=0 THEN 160ELSE ZP%=1:GOSUB 7700
160 ZQ%=0:PRINT"*SAVE    issued    COMMANDS";:GOSUB 5820
170 IF JJ=0 THEN 210ELSE ZQ%=1
180 OPEN "O",2,"IMAM2COM.DAT":GOTO 210
190 IF ERR=7 THEN 8530 ELSE EE%=9:RESUME 680
200 'Main Menu - Task Group Selection
210 GOSUB 8480:PRINT "I M A M   Main Menu:":PP%=0:EE%=0:LL=9
220 FOR KK=0 TO LL:GOSUB 7160
230 GOSUB 7830
240 NEXT KK
250 GOTO 550
260 'Input/Output/Storage Management Command Selection
270 LL=9:GOSUB 7800
280 FOR KK=1 TO LL:GOSUB 5280
290 GOSUB 7830
300 NEXT KK
310 GOTO 550
320 'Assignment Statement
330 LL=20:GOSUB 7800
340 FOR I=1 TO 10
350 KK=I:GOSUB 4960:PRINT:PRINT TAB(10);KK;" -    ";Q$;
360 KK=I+10:GOSUB 4960:PRINT TAB(40);KK;" -    ";Q$
370 NEXT I
380 GOTO 550
390 'Arithmetic Statements Selection
400 LL=9:GOSUB 7800
410 FOR KK=1 TO LL:GOSUB 5180
420 GOSUB 7830
430 NEXT KK:GOTO 550
440 'Special Matrix Operations Selection
450 LL=8:GOSUB 7800
460 FOR KK=1 TO LL:GOSUB 5390
470 GOSUB 7830
480 NEXT KK:GOTO 550
490 'Utilities Command Selection
500 LL=4:GOSUB 7800
510 FOR KK=1 TO LL:GOSUB 5490
520 GOSUB 7830
530 NEXT KK:GOTO 550
540 'Command Selection Routine
550 PRINT:PRINT TAB(5)"Enter SELECTION ";:IF PP%=0 THEN 570
560 PRINT" (or 0 to return to MAIN MENU) ";
570 R$="":LINE INPUT R$:IF R$="" THEN 210 ELSE KK=VAL(R$)
```

```
580 IF KK=0 THEN 8060 ELSE IF KK>LL THFN 550
590 JJ=1:IF PP%>0 THEN 660ELSE IF KK>0 AND KK<6 THEN 600 ELSE GOSUB 8480:GOSUB 7
960
600 IF JJ=0 THEN 210 ELSE PP%=KK:IF KK=0 THEN 7990 ELSE ON KK GOTO 270,330,400,4
50,500,7930,610,7260,7260
610 'User Module Processing
620 PQ%=1:GOTO 10000
630 'User Entry Point
640 IF PP%>0 AND PP%<6 THEN 660ELSE PRINT:PRINT"**** INVALID CALL FROM USER MODU
LE.":RETURN
650 'Switch for Command Groups
660 ON PP% GOTO 1060,1520,2160,2780,860
670 'Error Messages
680 IF PQ%>0 THEN RETURN ELSE PRINT:ON EE% GOTO 690,700,710,720,730,740,750,760,
770,780,790,800
690 R$="Unable to Allocate Storage for "+Q$:GOTO 820
700 R$="Matrix "+Q$+" is a NULL matrix":GOTO 82C
710 R$="Invalid Pivot Index for "+Q$:GOTO 820
720 R$="Invalid Submatrix Size Specification":GOTO 820
730 R$="Unable to Invert [ A ]":GOTO 820
740 R$="Matrix "+Q$+" Not Square":GOTO 820
750 R$="[ B ] and [ A ] Not Conformable":GOTO 820
760 R$="[ B ] and [ C ] Not Conformable":GOTO 820
770 R$="Operation NOT EXECUTAPLE":GOTO 820
780 PRINT:PRINT "**** ERROR";ERR;:R$="Operation ABORTED":GOTO 820
790 R$="An INVALID HEADER RECORD was READ":GOTO 820
800 R$="Solution of Simultaneous Equations Unsuccessful":GOTO 820
810 R$="SYSTEM ERROR":GOTO 820
820 R$="**** "+R$:EE%=0:IF ZP%>0 THEN LPRINT R$
830 IF ZQ%>0 THEN PRINT#2,R$
840 PRINT:PRINT R$:GOTO 7500
850 'Utilities Command Execution
860 GOSUB 7750
870 IF JJ=0 THEN 8560
880 ON KK GOTO 7850,890,900,4440
890 IF ZQ%=0 THEN 7480ELSE OPEN "O",3,"IMAM3COM.DAT":CLOSE 2:GOTO 6020
900 IF FK%=0 THEN 7480ELSE NN=1:GOSUB 7680
910 IF IP=0 THEN 930
920 LPRINT:LPRINT"FILE DIRECTORY:":LPRINT TAB(17);"ROWS";TAB(27);"COLS";TAB(50);
"IDENTIFICATION":LPRINT
930 JJ=0:GOSUB 8480:PRINT TAB(17);"ROWS";TAB(27);"COLS";TAB(50);"IDENTIFICATION"
:PRINT
940 XR%=NN:GOSUB 7590
950 PRINT TAB(10) USING"     #####";I;J;:PRINT TAB(40);LEFT$(T$,36):PRINT TAB(40
);MID$(T$,37,36):PRINT:IF IP=0 THEN 970
960 LPRINT TAB(10) USING"     #####";I;J;:LPRINT TAB(40);LEFT$(T$,36):LPRINT TAB
(40);MID$(T$,37,36)
970 IF FK%=NN THEN 7490ELSE JJ=JJ+2
980 IF JJ<20 THEN 940
990 IF INKEY$="" THEN 990ELSE 930
1000 'Reread issued commands
1010 CLOSE 2:CLOSE 3:OPEN "I",3,"IMAM3COM.DAT":OPEN "O",2,"IMAM2COM.DAT"
1020 INPUT #3,T$:IF EOF(3) THEN 1040
1030 PRINT #2,T$:GOTO 1020
1040 CLOSE 3:GOTO 7490
1050 'Input/Output/File Management
1060 GOSUB 7750
1070 IF JJ=0 THEN 8560
1080 IF KK=3 OR KK=4 THEN 1300
1090 MM=0
1100 ON KK GOTO 1110,1110,1300,1300,1390,1390,1460,3800,3800
1110 IF KK=1 THEN II=0 ELSE II=1
1120 GOSUB 5900
1130 IF EE%>0 THEN 210
1140 IF KK=1 THEN 1260
1150 II=0
1160 II=II+1:PRINT:PRINT"*Enter Next Element (No. ";II;"):"
1170 INPUT"*Row    Index";I:IF I>0 AND I<=NR THEN 1190ELSE GOSUB 7340
1180 IF JJ=1 THEN 1170ELSE 1240
1190 INPUT"*Column Index";J:IF J>0 AND J<=NC THEN 1210ELSE GOSUB 7340
1200 IF JJ=0 THEN 1240ELSE 1190
1210 INPUT"*Value";A(I,J)
1220 PRINT:PRINT"*MORE?":GOSUB 5830
1230 IF JJ=1 THEN 1160ELSE 1250
1240 II=II-1
1250 PRINT:PRINT"*Number of Elements Input";II:GOTO 7490
1260 PRINT:PRINT "*Input Elements of A by Rows:"
1270 FOR I=1 TO NR
1280 FOR J=1 TO NC:PRINT "A(";I;",";J;")=";:INPUT A(I,J)
1290 NEXT J,I:GOTO 7490
1300 IF KK=3 THEN MM=0 ELSE MM=8
1310 GOSUB 4890
1320 IF EE%>0 THEN 680ELSE GOSUB 4940
1330 GOSUB 8480:PRINT "*Current Elements of Matrix ";Q$
1340 FOR I=1 TO NR:PRINT:GOSUB 6150
1350 PRINT TAB(JJ);R$
1360 FOR J=1 TO NC:IF MM=0 THEN PRINT A(I,J), ELSE PRINT S(I,J),
1370 NEXT J,I:PRINT:IF PQ%=0 THEN 7490 ELSE 8520
1380 'print A or S
1390 GOSUB 4890
1400 IF EE%>0 THEN 680
1410 GOSUB 7700
1420 FOR I=1 TO NR:LPRINT:GOSUB 6150
1430 LPRINT TAB(JJ)R$:JJ=0
1440 FOR J=1 TO NC:JJ=JJ+1:IF JJ<=5 THEN 1445 ELSE JJ=1:LPRINT
1445 IF KK=5 THEN LPRINT A(I,J), ELSE LPRINT S(I,J),
1450 NEXT J,I:LPRINT:GOTO 7490
1460 FOR MM=0 TO 8:IF TT%(0,MM)=0 THEN 1500ELSE GOSUB 4940
1470 PRINT"*Erase ";Q$;:GOSUB 5820
1480 IF JJ=0 THEN 1500ELSE GOSUB 4600
1490 PRINT TAB(30) Q$;" ERASED."
1500 NEXT MM:GOTO 7490
1510 'Assignment Statements
1520 GOSUB 7750
1530 IF JJ=0 THEN 8560
1540 IF KK<>1 AND KK<>11 THEN 1580
1550 IF KK=1 THEN MM=0 ELSE MM=8
1560 II=0:GOSUB 5900
1570 IF EE%=0 THEN 7490ELSE 680
1580 IF KK=10 OR KK=20 THEN 1930ELSE IF KK>11 THEN 1760
1590 MM=0:GOSUB 4890
1600 IF EE%>0 THEN 680
```

```
1610 MM=KK-1:GOSUB 4720
1620 IF EE%>0 THEN 680
1630 FOR I=1 TO NR
1640 FOR J=1 TO NC
1650 ON KK-1 GOTO 1660,1670,1680,1690,1700,1710,1720,1730
1660 B(I-1,J-1)=A(I,J):GOTO 1740
1670 C(I-1,J-1)=A(I,J):GOTO 1740
1680 D(I-1,J-1)=A(I,J):GOTO 1740
1690 E(I-1,J-1)=A(I,J):GOTO 1740
1700 F(I-1,J-1)=A(I,J):GOTO 1740
1710 G(I-1,J-1)=A(I,J):GOTO 1740
1720 H(I-1,J-1)=A(I,J):GOTO 1740
1730 S(I,J)=A(I,J)
1740 NEXT J,I
1750 GOTO 7490
1760 MM=KK-11:GOSUB 4890
1770 IF EE%>0 THEN 680
1780 MM=0:GOSUB 4720
1790 IF EE%>0 THEN 680
1800 FOR I=1 TO NP
1810 FOR J=1 TO NC
1820 ON KK-11 GOTO 1830,1840,1850,1860,1870,1880,1890,1900
1830 A(I,J)=B(I-1,J-1):GOTO 1910
1840 A(I,J)=C(I-1,J-1):GOTO 1910
1850 A(I,J)=D(I-1,J-1):GOTO 1910
1860 A(I,J)=E(I-1,J-1):GOTO 1910
1870 A(I,J)=F(I-1,J-1):GOTO 1910
1880 A(I,J)=G(I-1,J-1):GOTO 1910
1890 A(I,J)=H(I-1,J-1):GOTO 1910
1900 A(I,J)=S(I,J)
1910 NEXT J,I
1920 GOTO 7490
1930 IP=IX(1):JP=IX(2):MM=0:GOSUB 4890
1940 IF EE%=0 THEN 2020
1950 IF PQ%>0 THEN RETURN
1960 IF KK=20 THEN 1990
1970 PRINT:PRINT"[ A ] is NULL.":PRINT"Is it to INITIALIZE a SUBMATRIX of [ B ]"
;:GOSUB 5820
1980 IF JJ=0 THEN 680
1990 II=0:GOSUB 5900
2000 IF EE%>0 THEN 680
2010 IB=NR:JB=NC:GOTO 2030
2020 IF KK=10 THEN 2010ELSE 1990
2030 MM=8:IF KK=10 THEN 2050ELSE GOSUB 4890
2040 IF EE%>0 THEN 680ELSE 2070
2050 II=1:GOSUB 5900
2060 IF EE%>0 THEN 680
2070 JJ=0:LL=NR:NN=NC:GOSUB 5550
2080 IF EE%>0 THEN 210
2090 FOR I=1 TO IB:II=IP+I-1
2100 FOR J=1 TO JB:JJ=JP+J-1
2110 IF KK=10 THEN S(II,JJ)=A(I,J) ELSE A(I,J)=S(II,JJ)
2120 NEXT J,I
2130 GOSUB 6240
2140 GOTO 7490
2150 'Arithmetic Statements
2160 GOSUB 7750
2170 IF JJ=0 THEN 8560
2180 IF KK>4 THEN 2470
2190 MM=1:GOSUB 4890
2200 IF EE%>0 THEN 680ELSE IF KK=4 THEN 2420
2210 IB=NR:JB=NC:MM=2:GOSUB 4890
2220 IF EE%>0 THEN 680
2230 IC=NR:JC=NC:MM=0
2240 ON KK GOTO 2340,2250,2250
2250 IF IB=IC AND JB=JC THEN 2270
2260 EE%=8:GOTO 680
2270 GOSUB 4720
2280 IF EE%>0 THEN 680
2290 FOR I=1 TO NR
2300 FOR J=1 TO NC
2310 IF KK=2 THEN A(I,J)=B(I-1,J-1)+C(I-1,J-1) ELSE A(I,J)=B(I-1,J-1)-C(I-1,J-1)
2320 NEXT J,I
2330 GOTO 7490
2340 IF JB<>IC THEN 2260
2350 NR=IB:NC=JC:GOSUB 4720
2360 IF EE%>0 THEN 680
2370 FOR I=0 TO NR-1
2380 FOR J=0 TO NC-1:X=0!
2390 FOR K=0 TO JB-1:X=X+B(I,K)*C(K,J):NEXT K
2400 A(I+1,J+1)=X:NEXT J,I
2410 GOTO 7490
2420 SWAP NR,NC:MM=0:GOSUB 4720
2430 IF EE%>0 THEN 680
2440 FOR I=1 TO NR
2450 FOR J=1 TO NC:A(I,J)=B(J-1,I-1):NEXT J,I
2460 GOTO 7490
2470 MM=0:GOSUB 4890
2480 IF EE%>0 THEN 680
2490 IF KK>6 THEN 2610
2500 IF KK=6 THEN 2560
2510 IF PQ%=0 THEN 2520ELSE X=A(0,0):GOTO 2560
2520 PRINT:INPUT "*Value of Scalar";X
2530 Q$="Value of Scalar "+STR$(X)
2540 IF ZP%>0 THEN LPRINT Q$
2550 IF ZQ%>0 THEN PRINT #2,Q$
2560 FOR I=1 TO NR
2570 FOR J=1 TO NC
2580 IF KK=5 THEN A(I,J)=A(I,J)*X ELSE A(I,J)=-A(I,J)
2590 NEXT J,I
2600 GOTO 7490
2610 IF KK=8 THEN 2670
2620 IF NR=NC THEN 2640
2630 EE%=6:GOTO 680
2640 N=NR:IF KK=9 THEN GOSUB 8080 ELSE GOSUB 6320
2650 IF EE%>0 THEN 680
2660 PRINT:PRINT "*Determinant = ";X:GOTO 7490
2670 M=NC:N=NR:MM=8:GOSUB 4890
2680 IF EE%>0 THEN 680
2690 IF N=NR THEN 2710
2700 EE%=7:GOTO 680
2710 IF NR<>NC THEN 2630
```

```
2720 GOSUB 6730
2730 MM=8:GOSUB 4590
2740 IF EE%>0 THEN 680 ELSE 7490
2770 'Special Matrix Operations
2780 GOSUB 7750
2790 IF JJ=0 THEN 8560
2800 IF KK>4 THEN 3290ELSE MM=0
2810 GOSUB 4890
2820 IF EE%>0 THEN 680
2830 IF PQ%>0 THEN 3200
2840 IF KK>2 THEN 2860
2850 R$=" row ":LL=NR:NN=NC:GOTO 2870
2860 R$=" column ":LL=NC:NN=NR
2870 ON KK GOTO 2880,2890,2880,2890
2880 T$="*From":Q$=" to ":GOTO 2900
2890 T$="*To":Q$=" from "
2900 FOR I=1 TO LL:IF KK<3 THEN A(I,0)=0 ELSE A(0,I)=0
2910 NEXT I
2920 FOR I=1 TO LL:PRINT:PRINT T$;R$;I;Q$;:INPUT II
2930 IF II>=1 AND II<=LL THEN 2960
2940 PRINT:PRINT TAB(10)"***** Invalid";R$;"number. Please restart."
2950 GOTO 2900
2960 IF KK=2 THEN 3060
2970 IF KK=4 THEN 3070ELSE J=1
2980 IF I=J THEN 3030
2990 IF KK=3 THEN 3010
3000 IF A(J,0)=II THEN 2940ELSE 3020
3010 IF A(0,J)=II THEN 2940
3020 J=J+1:GOTO 2980
3030 IF KK=3 THEN 3050
3040 A(I,0)=II:GOTO 3080
3050 A(0,I)=II:GOTO 3080
3060 IF A(II,0)>0 THEN 2940ELSE A(II,0)=I:GOTO 3080
3070 IF A(0,II)>0 THEN 2940ELSE A(0,II)=I
3080 NEXT I
3090 IF KK<3 THEN R$=" ROW " ELSE R$=" COLUMN "
3100 T$="*Specified"+R$+"Permutation:":PRINT:PRINT T$:IF ZP%>0 THEN LPRINT T$
3110 IF ZQ%>0 THEN PRINT#2,T$
3120 PRINT:IF KK=1 OR KK=3 THEN R$=" --> " ELSE R$=" <-- "
3130 FOR I=1 TO LL:PRINT I;R$;:IF KK<3 THEN PRINT A(I,0), ELSE PRINT A(0,I),
3140 IF ZP%=0 THEN 3160
3150 LPRINT I;R$;:IF KK<3 THEN LPRINT A(I,0), ELSE LPRINT A(0,I),
3160 IF ZQ%=0 THEN 3170ELSE IF KK<3 THEN PRINT #2,A(I,0) ELSE PRINT #2,A(0,I)
3170 NEXT I:PRINT:IF ZP%=1 THEN LPRINT
3180 GOSUB 7420
3190 IF JJ=0 THEN 7470
3200 FOR I=1 TO LL
3210 IF KK>2 THEN II=A(0,I) ELSE II=A(I,0)
3220 IF II=I THEN 3270
3230 FOR J=0 TO NN:IF KK>2 THEN 3250
3240 SWAP A(I,J),A(II,J):GOTO 3260
3250 SWAP A(J,I),A(J,II)
3260 NEXT J
3270 NEXT I
3280 GOTO 7490
3290 IF KK>6 THEN 3560
3300 MM=1:GOSUB 4890
3310 IF EE%>0 THEN 680
3320 IB=NR:JB=NC:MM=2:GOSUB 4890
3330 IF EE%>0 THEN 680
3340 IC=NR:JC=NC:IF JC=IC THEN 3360
3350 EE%=6:GOTO 680
3360 IF JC=IB THEN 3370ELSE EE%=8:GOTO 680
3370 IF KK=5 THEN 3420ELSE MM=8:II=1:GOSUB 5900
3380 IF EE%>0 THEN 680
3390 LL=NR:NN=NC:IQ=JB:SWAP IQ,IB:JJ=0:GOSUB 5550
3400 IF EE%>0 THEN 680
3410 SWAP IQ,IB:GOTO 3440
3420 MM=0:NR=JB:NC=JB:GOSUB 4720
3430 IF EE%>0 THEN 680
3440 FOR J=0 TO JB-1
3450 FOR I=0 TO JB-1:X=0!
3460 FOR JJ=0 TO IB-1:XX=0!
3470 FOR II=0 TO IB-1
3480 XX=XX+C(II,JJ)*B(II,I):NEXT II
3490 X=X+B(JJ,J)*XX:NEXT JJ
3500 IF KK=5 THEN 3520
3510 S(I+IP,J+JP)=S(I+IP,J+JP)+X:GOTO 3530
3520 A(I+1,J+1)=X
3530 NEXT I,J
3540 GOSUB 6240
3550 GOTO 7490
3560 MM=8:II=1:GOSUB 5900
3570 IF EE%>0 THEN 680
3580 JJ=0:LL=NR:NN=NC:IB=1:JB=1:GOSUB 5550
3590 IF EE%>0 THEN 210ELSE MM=0:GOSUB 4890
3600 IQ=IP:JQ=JP:SWAP LL,NR:SWAP NN,NC
3610 JJ=2:GOSUB 5550
3620 IF EE%>0 THEN 210
3630 SWAP LL,NR:SWAP NN,NC
3640 GOSUB 5690
3650 IF EE%>0 THEN 210
3660 IF IQ+IC-1<=LL THEN 3690ELSE GOSUB 7340
3670 IF JJ=1 THEN 3640
3680 GOTO 210
3690 IF JQ+JC-1<=NN THEN 3710ELSE GOSUB 7340
3700 IF JJ=1 THEN 3640ELSE 3680
3710 FOR I=1 TO IC:II=IQ+I-1:LL=IP+I-1
3720 FOR J=1 TO JC:JJ=JQ+J-1:NN=JP+J-1
3730 IF KK=7 THEN 3750
3740 S(II,JJ)=A(LL,NN):GOTO 3760
3750 S(II,JJ)=S(II,JJ)+A(LL,NN)
3760 NEXT J,I
3770 GOSUB 6250
3780 GOTO 7490
3790 'File Management
3800 IF FK%=0 THEN 7480ELSE XR%=FK%
3810 PRINT:PRINT "PRESS:":PRINT:PRINT TAB(10) "< 1 > To TAG [ A ]":PRINT:PRINT
TAB(10) "< 2 > To TAG [ B ]"
3820 R$=INKEY$:IF R$="" THEN 3820
3830 IF R$="1" THEN 3860 ELSE IF R$="2" THEN 3870 ELSE 3820
3860 MM=0:GOTO 3880
```

```
3870 MM=8
3880 IF KK=9 THEN 4330ELSE GOSUB 4890
3890 IF EE%>0 THEN 680
3900 PRINT:PRINT "*Allocate a New File";:GOSUB 5820
3910 IF JJ=1 THEN 3920ELSE 4100
3920 GOSUB 5860
3930 IC=IC+XR%
3940 PRINT:INPUT"*Full ID:";T$
3950 II=XR%:GOSUB 6220
3990 JB=80
4000 FOR J=1 TO NC
4010 FOR I=1 TO NR:IF JB=80 THEN GOSUB 7890
4020 JB=JB+4
4030 IF MM=0 THEN T$=T$+MKS$(A(I,J)) ELSE T$=T$+MKS$(S(I,J))
4040 NEXT I,J:IF JB=0 THEN 4070ELSE GOSUB 7890
4070 IF FK%>=IC THEN 7490ELSE FK%=IC:GOSUB 7490
4090 'Look for old file of adequate size
4100 XR%=1
4110 GOSUB 7590
4120 IF NR*NC>20*(NN-XR%-1) THEN 4150ELSE GOSUB 6100
4130 PRINT"*O.K. to USE this File";:GOSUB 5820
4140 IF JJ=0 THEN 4150ELSE IC=NN:GOTO 3940
4150 XR%=NN:IF XR%<FK% THEN 4110
4160 PRINT:PRINT "*No Available Slot. Must Use NEW FILE":GOSUB 5830
4170 IF JJ=0 THEN 210ELSE 3920
4220 NR=I:NC=J
4230 GOSUB 8480:PRINT "ID: ";T$:PRINT "Number of Rows    = ";I:PRINT "Number of
Columns = ";J:GOSUB 4720
4240 IF EE%>0 THEN 680ELSE IB=81
4250 FOR J=1 TO NC
4260 FOR I=1 TO NR:IF IB<81 THEN 4280
4270 XR%=XR%+1:GET 1,XR%:IB=1
4280 X=CVS(MID$(BF$,IB,4))
4290 IF MM=0 THEN A(I,J)=X ELSE S(I.J)=X
4300 IB=IB+4:NEXT I,J
4310 GOTO 7490
4320 'Search for needed file
4330 XR%=1
4340 SD$="":PRINT:INPUT "*Search ID";SD$
4350 JP=LEN(SD$)
4360 GOSUB 7590
4380 IF SD$<>LEFT$(T$,JP) THEN 4410ELSE 4220
4410 XR%=NN:IF NN<FK% THEN 4360
4420 PRINT:PRINT"[ ";SD$;" ] NOT in FILE":IF PQ%=2 THEN RETURN ELSE 7500
4430 'Garbage Disposal
4440 IF FK%=0 THEN 7480
4450 IP=1:XR%=1
4460 GOSUB 6180
4470 PRINT:PRINT "*OK to DELETE?":GOSUB 5830
4480 IF JJ=1 THEN 4540 ELSE NR=I:NC=J:GOSUB 5860
4490 IC=IP+IC
4500 T$=LEFT$(BF$,6)+MKI$(IC)+T$:LSET BF$=T$:PUT 1,IP
4510 IF IP<XR% THEN 4530 ELSE IP=IC:GOTO 4540
4520 GET 1,XR%:PUT 1,IP
4530 IP=IP+1:IF IP=IC THEN 4540 ELSE XR%=XR%+1:GOTO 4520
4540 XR%=NN
4550 IF XR%<FK% THEN 4460
4560 FK%=IC:GOTO 7490
4570 'STORAGE MANAGEMENT PRIMITIVES
4580 'Erase a matrix
4590 II=TT%(0,MM):IF II=0 THEN RETURN ELSE JJ=TT%(1,MM):IF MM<>0 AND MM<>8 THEN
4600 ELSE II=II+1:JJ=JJ+1
4600 TT%(0,MM)=0:TT%(1,MM)=0:AA%=AA%-II*JJ
4610 ON MM+1 GOTO 4620,4630,4640,4650,4660,4670,4680,4690,4700
4620 ERASE A:RETURN
4630 ERASE B:RETURN
4640 ERASE C:RETURN
4650 ERASE D:RETURN
4660 ERASE E:RETURN
4670 ERASE F:RETURN
4680 ERASE G:RETURN
4690 ERASE H:RETURN
4700 ERASE S:RETURN
4710 'Allocate a Matrix
4720 IF TT%(0,MM)>0 THEN GOSUB 4590
4730 II=NR:JJ=NC
4740 IF MM=0 OR MM=8 THEN 4760
4750 II=II-1:JJ=JJ-1
4760 AA%=AA%+(II+1)*(JJ+1)
4770 TT%(0,MM)=NR:TT%(1,MM)=NC
4780 ON MM+1 GOTO 4790,4800,4810,4820,4830,4840,4850,4860,4870
4790 DIM A(II,JJ):RETURN
4800 DIM B(II,JJ):RETURN
4810 DIM C(II,JJ):RETURN
4820 DIM D(II,JJ):RETURN
4830 DIM E(II,JJ):RETURN
4840 DIM F(II,JJ):RETURN
4850 DIM G(II,JJ):RETURN
4860 DIM H(II,JJ):RETURN
4870 DIM S(II,JJ):RETURN
4880 'Get size of matrix
4890 GOSUB 4940
4900 IF TT%(0,MM)>0 THEN 4920
4910 EE%=2:RETURN
4920 NR=TT%(0,MM):NC=TT%(1,MM):RETURN
4930 'Get matrix identifier
4940 Q$=MID$("ABCDEFGHS",MM+1,1):Q$="[  "+Q$+"  ]":RETURN
4950 'Assignment Command Images
4960 ON KK GOTO 4970,4980,4990,5000,5010,5020,5030,5040,5050,5060,5070,5080,5090
,5100,5110,5120,5130,5140,5150,5160
4970 Q$="[0] --> A":RETURN
4980 Q$="A --> B":RETURN
4990 Q$="A --> C":RETURN
5000 Q$="A --> D":RETURN
5010 Q$="A --> E":RETURN
5020 Q$="A --> F":RETURN
5030 Q$="A --> G":RETURN
5040 Q$="A --> H":RETURN
5050 Q$="A --> S":RETURN
5060 Q$="A --> S[p;q]":RETURN
5070 Q$="S <-- [0]":RETURN
5080 Q$="A <-- B":RETURN
```

```
5090 Q$="A <-- C":RETURN
5100 Q$="A <-- D":RETURN
5110 Q$="A <-- E":RETURN
5120 Q$="A <-- F":RETURN
5130 Q$="A <-- G":RETURN
5140 Q$="A <-- H":RETURN
5150 Q$="A <-- S":RETURN
5160 Q$="A <-- S[p;q]":RETURN
5170 'Arithmetic Statement Command Images
5180 ON KK GOTO 5190,5200,5210,5220,5230,5240,5250,5260,8070
5190 Q$="A <-- B * C":RETURN
5200 Q$="A <-- B + C":RETURN
5210 Q$="A <-- B - C":RETURN
5220 Q$="A <-- Transpose of B":RETURN
5230 Q$="A <-- c * A (c a scalar)":RETURN
5240 Q$="A <-- Negative  of A":RETURN
5250 Q$="A <-- Inverse   of A":RETURN
5260 Q$="SOLVE:  S X = A  for X (RESULT --> A)":RETURN
5270 'Input/output Command Images
5280 ON KK GOTO 5290,5300,5310,5320,5330,5340,5350,5360,5370
5290 Q$="Input Elements of A (by Rows)":RETURN
5300 Q$="Input Selected Elements of A":RETURN
5310 Q$="Display A":RETURN
5320 Q$="Display S":RETURN
5330 Q$="Print A":RETURN
5340 Q$="Print S":RETURN
5350 Q$="ERASE Selected Matrices  (A;...;H;S)":RETURN
5360 Q$="A or S --> FILE":RETURN
5370 Q$="A or S <-- FILE":RETURN
5380 'Special Matrix Arithmetic Operations
5390 ON KK GOTO 5400,5410,5420,5430,5440,5450,5460,5470
5400 Q$="'TO'   Row Permutation      A(I;.)     --> A(P(I);.)":RETURN
5410 Q$="'FROM' Row Permutation      A(P(I);.) --> A(I;.)":RETURN
5420 Q$="'TO'   Column Permutation  A(.;I)     --> A(.;P(I))":RETURN
5430 Q$="'FROM' Column Permutation  A(.;P(I)) --> A(.;I)":RETURN
5440 Q$="A        <-- B' * C * B":RETURN
5450 Q$="S[p;q] <-- S[p;q] + B' * C * B":RETURN
5460 Q$="S[p;q] <-- S[p;q] + A[r;s:nr;ns]":RETURN
5470 Q$="S[p;q] <-- A[r;s:nr;ns]":RETURN
5480 'Utilities
5490 ON KK GOTO 5500,5510,5520;5530
5500 Q$="DISPLAY Current Sizes of Matrices":RETURN
5510 Q$="DISPLAY/LIST Commands Executed in Current Run":RETURN
5520 Q$="DISPLAY/LIST Directory of Filed Matrices":RETURN
5530 Q$="COMPACT the file (Garbage Collection).":RETURN
5540 'Input Indices of Pivotal element
5550 IF PQ%=0 THEN 5570
5560 IP=IX(JJ+1):JP=IX(JJ+2):RETURN
5570 GOSUB 4940
5580 PRINT:PRINT"*Input the INDICES of the PIVOTAL ELEMENT of ";Q$
5590 INPUT "      Row     Index";IP:IF IP>0 THEN 5620ELSE GOSUB 7340
5600 IF JJ=1 THEN 5590
5610 EE%=3:RETURN
5620 IF IP+IB-1<=LL THEN 5640ELSE GOSUB 7340
5630 IF JJ=1 THEN 5590ELSE 5610
5640 INPUT "      Column Index";JP:IF JP>0 THEN 5660ELSE GOSUB 7340
5650 IF JJ=1 THEN 5640ELSE 5610
5660 IF JP+JB-1<=NN THEN RETURN ELSE GOSUB 7340
5670 IF JJ=1 THEN 5640ELSE 5610
5680 'Size of submatrix
5690 IF PQ%=0 THEN 5710
5700 IC=IX(5):JC=IX(6):RETURN
5710 PRINT:PRINT"*Size of Submatrix of ";Q$;":"
5720 INPUT "*Number  of   Rows";IC:IF IC>0 THEN 5750ELSE GOSUB 7340
5730 IF JJ=1 THEN 5720
5740 EE%=4:RETURN
5750 IF IP+IC-1<=NR THEN 5770ELSE GOSUB 7340
5760 IF JJ=1 THEN 5720ELSE 5740
5770 INPUT "*Number of Columns";JC:IF JC>0 THEN 5790ELSE GOSUB 7340
5780 IF JJ=1 THEN 5770ELSE 5740
5790 IF JP+JC-1<=NC THEN RETURN ELSE GOSUB 7340
5800 IF JJ=1 THEN 5770ELSE 5740
5810 'Query
5820 PRINT"?"
5830 PRINT"*Press < 1 > if YES, < 0 > if NO":JJ=0
5840 R$=INKEY$:IF R$="" THEN 5840 ELSE IF R$="0" THEN RETURN
5850 IF R$<>"1" THEN 5830 ELSE JJ=1:RETURN
5860 IC=NR*NC/20:IF 20*IC<NR*NC THEN 5880
5870 IC=IC+1:RETURN
5880 IC=IC+2:RETURN
5890 'Establish or change size of matrix
5900 GOSUB 7300
5910 IF EE%>0 THEN 5930
5920 IF II>0 THEN 5990ELSE 5940
5930 IF II=0 THEN 5960ELSE 5990
5940 IF PQ%>0 THEN RETURN ELSE PRINT"*Change Size?":GOSUB 5830
5950 IF JJ=0 THEN 5970
5960 EE%=0:GOSUB 7530
5970 IF EE%>0 THEN RETURN ELSE GOSUB 4720
5980 RETURN
5990 IF PQ%>0 THEN RETURN ELSE PRINT"*Initialize?":GOSUB 5830
6000 IF JJ=1 THEN 5960ELSE RETURN
6010 'Display IMAM Commands issued
6020 CLOSE 2:OPEN "I",2,"IMAM2COM.DAT"
6030 SD$="IMAM Version 2. COMMANDS Issued:":GOSUB 7680
6040 IF IP=0 THEN 6050ELSE LPRINT:LPRINT SD$:LPRINT
6050 JJ=0:GOSUB 8480:PRINT SD$:PRINT
6060 JJ=JJ+1:INPUT #2,R$:IF EOF(2) THEN 6110ELSE PRINT TAB(5)">";R$
6070 IF IP=1 THEN LPRINT TAB(5)">";R$
6080 IF PP%>0 THEN PRINT #3,R$
6090 IF JJ<20 THEN 6060
6100 IF INKEY$="" THEN 6100ELSE 6050
6110 IF PP%>0 THEN 1010ELSE SD$="END of RUN."
6120 IF IP=1 THEN LPRINT TAB(5)">";SD$
6130 RETURN
6140 'Matrix Row Header
6150 IF NC<5 THEN JJ=8*NC-7 ELSE JJ=32
6160 R$="ROW [ "+RIGHT$(STR$(I),LEN(STR$(I))-1)+" ]:":RETURN
6170 'Get and Display Header Data
6180 GOSUB 7590
6190 IF EE%>0 THEN RETURN
6200 PRINT:PRINT"*ID:";T$:PRINT TAB(8)"Number  of   Rows";I:PRINT TAB(8)"Number
of Columns";J:RETURN
```

```
6230 'Record pivot indices
6240 IF ZQ%=0 THEN RETURN ELSE PRINT #2,"Indices p;q "+STR$(IP)+STR$(JP):RETURN
6250 IF ZQ%=0 THEN RETURN ELSE PRINT #2,"Indices p;q;r;s "+STR$(IQ)+STR$(JQ)+STR
$(IP)+STR$(JP)
6260 PRINT#2"Submatrix size "+STR$(IC)+" by "+STR$(JC):RETURN
6310 'Matrix Inversion
6320 X=1'
6330 FOR II=1 TO N:A(0,II)=II:A(II,0)=II:NEXT II
6340 FOR JJ=1 TO N:IF JJ=N THEN 6440
6350 YY=0:LL=JJ:MM=JJ
6360 FOR II=JJ TO N
6370 FOR KK=JJ TO N
6380 QQ=ABS(A(II,KK)):IF QQ<=YY THEN 6400
6390 YY=QQ:LL=II:MM=KK
6400 NEXT KK:NEXT II:IF LL=JJ THEN 6420
6410 FOR II=0 TO N:SWAP A(JJ,II),A(LL,II):NEXT II
6420 IF MM=JJ THEN 6440
6430 FOR II=0 TO N:SWAP A(II,JJ),A(II,MM):NEXT II
6440 QQ=A(JJ,JJ):IF QQ<>0 THEN 6460
6450 X=0':EE%=5:RETURN
6460 FOR II=1 TO N:IF II=JJ THEN 6480
6470 A(JJ,II)=A(JJ,II)/QQ:GOTO 6490
6480 A(JJ,JJ)=1'/QQ:X=X*QQ
6490 NEXT II
6500 FOR II=1 TO N:IF II=JJ THEN 6560ELSE QQ=A(II,JJ)
6510 FOR KK=1 TO N
6520 IF KK=JJ THEN 6540ELSE A(II,KK)=A(II,KK)-A(JJ,KK)*QQ
6530 GOTO 6550
6540 A(II,JJ)=-QQ*A(JJ,JJ)
6550 NEXT KK
6560 NEXT II
6570 NEXT JJ
6580 FOR II=1 TO N:SWAP A(0,II),A(II,0):NEXT II
6590 FOR II=1 TO N-1:IF A(II,0)=II THEN 6640
6600 FOR JJ=II+1 TO N:IF A(JJ,0)<>II THEN 6630
6610 FOR KK=0 TO N:SWAP A(II,KK),A(JJ,KK):NEXT KK
6620 GOTO 6640
6630 NEXT JJ
6640 NEXT II
6650 FOR II=1 TO N-1:IF A(0,II)=II THEN 6700
6660 FOR JJ=II+1 TO N:IF A(0,JJ)<>II THEN 6690
6670 FOR KK=0 TO N:SWAP A(KK,II),A(KK,JJ):NEXT KK
6680 GOTO 6700
6690 NEXT JJ
6700 NEXT II
6710 RETURN
6720 'Simultaneous Equations Solver
6730 FOR II=1 TO N:S(0,II)=II:NEXT II
6740 FOR II=1 TO N:QQ=0:LL=II:MM=II
6750 FOR JJ=II TO N
6760 FOR KK=II TO N:IF QQ>=ABS(S(JJ,KK)) THEN 6780
6770 LL=JJ:MM=KK:QQ=ABS(S(JJ,KK))
6780 NEXT KK:NEXT JJ:IF II=LL THEN 6830
6790 FOR JJ=1 TO N
6800 QQ=S(II,JJ):S(II,JJ)=S(LL,JJ):S(LL,JJ)=QQ:NEXT JJ
6810 FOR JJ=1 TO M
6820 QQ=A(II,JJ):A(II,JJ)=A(LL,JJ):A(LL,JJ)=QQ:NEXT JJ
6830 IF II=MM THEN 6860
6840 FOR JJ=1 TO N
6850 QQ=S(JJ,II):S(JJ,II)=S(JJ,MM):S(JJ,MM)=QQ:NEXT JJ:SWAP S(0,II),S(0,MM)
6860 QQ=S(II,II):IF QQ<>0 THEN 6880
6870 EE%=12:RETURN
6880 QQ=1!/QQ
6890 IF II=N THEN 6920
6900 FOR JJ=II+1 TO N
6910 S(II,JJ)=S(II,JJ)*QQ:NEXT JJ
6920 FOR JJ=1 TO M
6930 A(II,JJ)=A(II,JJ)*QQ:NEXT JJ
6940 IF II<N THEN 6950ELSE 7000
6950 FOR JJ=II+1 TO N:QQ=S(JJ,II)
6960 FOR KK=II+1 TO N
6970 S(JJ,KK)=S(JJ,KK)-S(II,KK)*QQ:NEXT KK
6980 FOR KK=1 TO M:A(JJ,KK)=A(JJ,KK)-A(II,KK)*QQ:NEXT KK
6990 NEXT JJ
7000 NEXT II
7010 FOR II=N-1 TO 1 STEP -1
7020 FOR JJ=1 TO M
7030 FOR KK=II+1 TO N
7040 A(II,JJ)=A(II,JJ)-A(KK,JJ)*S(II,KK)
7050 NEXT KK
7060 NEXT JJ
7070 NEXT II
7080 FOR II=1 TO N:IF II=S(0,II) THEN 7130
7090 FOR JJ=1 TO N:LL=JJ:IF S(0,JJ)<>II THEN NEXT JJ ELSE 7100
7100 FOR KK=1 TO M
7110 QQ=A(II,KK):A(II,KK)=A(LL,KK):A(LL,KK)=QQ:NEXT KK
7120 S(0,LL)=S(0,II):S(0,II)=II
7130 NEXT II
7140 RETURN
7150 'Main Commands
7160 ON KK+1 GOTO 7980,7170,7180,7190,7200,7210,7220,7230,7240,7250
7170 Q$="Input/Output/File Management":RETURN
7180 Q$="Assignment (B;...;H;S to/from A)":RETURN
7190 Q$="Arithmetic Operation":RETURN
7200 Q$="Special Matrix Operation":RETURN
7210 Q$="Perform a Utility Function":RETURN
7220 Q$="Create a User Module":RETURN
7230 Q$="Enter a User Module":RETURN
7240 Q$="QUIT (End of Current Problem Run)":RETURN
7250 Q$="STOP (End of Session)":RETURN
7260 GOSUB 8480:PP%=0:IF ZQ%=0 THEN 7270 ELSE PRINT "Commands Issued Will Be Dis
played":GOSUB 6020
7270 IF FK%<=1 THEN 7280 ELSE GET 1,1:T$=LEFT$(BF$,4)+MKI$(FK%)+RIGHT$(BF$,74):L
SET BF$=T$:PUT 1,1
7280 IF KK=9 THEN SYSTEM ELSE PRINT "*Press ANY Key to START on a NEW PROBLEM"
7290 IF INKEY$="" THEN 7290 ELSE 140
7300 GOSUB 4890
7310 IF EE%=0 THEN 7320ELSE PRINT:PRINT Q$;"  IS NULL.":RETURN
7320 IF PQ%>0 THEN RETURN ELSE PRINT:PRINT Q$;"  has ";NR;" row(s) and ";NC;" co
lumn.s)":RETURN
7330 'Error condition handling and special routines
7340 PRINT:PRINT "*INVALID Input. RE-TRY?":GOTO 5830
7350 'Abort option
```

```
7360 IF PQ%=2 THEN 7390 ELSE PRINT:PRINT "*GO?"
7370 GOSUB 5830
7380 IF JJ=0 THEN RETURN ELSE PRINT "*TASK EXECUTION Ongoing."
7390 IF ZP%>0 THEN LPRINT TAB(5)">";Q$
7400 IF ZQ%>0 THEN PRINT #2,Q$
7410 RETURN
7420 PRINT:PRINT"*CONTINUE?":GOTO 5830
7460 'Set Error Codes
7470 EE%=10:GOTO 600
7480 EE%=9:GOTO 600
7490 IF PQ%=2 THEN RETURN ELSE PRINT:PRINT:PRINT "*Task COMPLETED."
7500 PRINT "*Press Any Key to CONTINUE."
7510 IF INKEY$="" THEN 7510ELSE IF PQ%>0 THEN RETURN ELSE 210
7520 GOSUB 4940
7530 PRINT:PRINT"*SIZE of Matrix ";Q$;":"
7540 INPUT"*Number of   Rows";NR:IF NR>0 THEN 7560ELSE GOSUB 7340
7550 IF JJ=1 THEN 7540ELSE EE%=10:RETURN
7560 INPUT"*Number of Columns";NC:IF NC>0 THEN RETURN ELSE GOSUB 7340
7570 IF JJ=1 THEN 7560ELSE EE%=10:RETURN
7580 'Get a Header
7590 GET 1,XR%
7600 I=CVI(MID$(BF$,1,2))
7610 J=CVI(MID$(BF$,3,2))
7630 NN=CVI(MID$(BF$,7,2))
7640 T$=MID$(BF$,9,72)
7660 RETURN
7670 'Line Printer Heading
7680 PRINT:PRINT"*Need a Hard Copy";:GOSUB 5820
7690 IP=0:IF JJ=0 THEN RETURN ELSE IP=1
7700 PRINT "*TOP of PAGE";:GOSUB 5820
7710 IF JJ=1 THEN LPHINT CHR$(12) ELSE LPRINT
7720 INPUT "*Title";T$
7730 LPRINT T$:RETURN
7740 'Command Display
7750 JJ=1:IF PQ%=2 THEN RETURN ELSE SWAP KK,PP%:GOSUB 7000
7760 SWAP KK,PP%:IF PP%=0 THEN 7010
7770 ON PP% GOSUB 5200,4960,5100,5390,5490
7780 PRINT"* T A S K : ";Q$:GOTO 7360
7790 'Task Group Display
7800 GOSUB 8480:GOSUB 7160
7810 PRINT Q$;":"
7820 RETURN
7830 PRINT:PRINT TAB(10);KK;" -   ";Q$:RETURN
7840 'Display Current Status of Matrices
7850 GOSUB, 8480
7860 FOR MM=0 TO 8:EE%=0:GOSUB 7300
7870 NEXT MM
7880 PRINT:PRINT AA%;" LOCATIONS Allocated for ARRAYS.":GOTO 7500
7890 LSET BF$=T$
7900 PUT 1,XR%:XR%=XR%+1:JB=0:T$="":RETURN
7910 PRINT:PRINT"**** INVALID FILE GENERATED ":GOTO 7490
7920 'User-Generated Modules
7930 GOSUB 8480:PRINT"**** KEY IN or MERGE your MODULES"
7940 PRINT TAB(5)"ENTRY POINT must be Statement Number Shown Below":PRINT TAB(5)
"Statement Numbers must be NOT LOWER than 10000."
7950 PRINT TAB(5)"RETURN to IMAM with a GOTO 210 Statement.":AUTO 10000:STOP
7960 GOSUB 7160
7970 GOTO 7780
7980 Q$="Line Print a Message":RETURN
7990 GOSUB 8480:PRINT "*TOP of PAGE?":GOSUB 5830
8000 IF JJ=1 THEN LPRINT CHR$(12) ELSE LPRINT
8010 GOSUB 8480:PRINT"*Key In MESSAGE (Limit Each LINE to 80 Characters)"
8020 PRINT"*Press ENTER After Each LINE.":PRINT"*Press ENTER TWICE After Keying
In LAST LINE."
8030 PRINT STRING$(79,61)
8040 Q$="":LINE INPUT Q$
8050 IF Q$="" THEN 210 ELSE LPRINT Q$:GOTO 8040
8060 IF PP%>0 THEN 210 ELSE 590
8070 Q$="Eigen-VP ([A]-e[B]): e's --> Diag [A], Vects --> Cols [B]":RETURN
8080 IF N>=2 THEN 8100
8090 R$="INVALID or NON-EXISTENT Data for Eigenvalue Problem.":GOTO 820
8100 MM=1:GOSUB 4890
8110 IF EE%>0 THEN 600
8120 IF NR<>NC OR NR<>N THEN 8090
8130 FOR I=0 TO N-1
8140 FOR J=I TO N-1:Q=B(J,I):IF I=0 THEN 8160
8150 FOR K=0 TO I-1:Q=Q-B(I,K)*B(J,K):NEXT K
8160 IF I<>J THEN 8190
8170 IF Q>0 THEN 8180 ELSE R$="[B] is Singular.":GOTO 820
8180 B(J,I)=SQR(Q):GOTO 8200
8190 B(J,I)=Q/B(I,I)
8200 NEXT J,I
8210 FOR J=0 TO N-1
8220 FOR I=0 TO J:IF J>I THEN 8230 ELSE B(J,J)=1/B(J,J):GOTO 8260
8230 Q=0
8240 FOR K=I TO J-1:Q=Q-B(J,K)*B(K,I):NEXT K
8250 B(J,I)=Q/B(J,J)
8260 NEXT I,J
8270 FOR I=0 TO N-2
8280 FOR J=I+1 TO N-1:B(I,J)=B(J,I):B(J,I)=0:NEXT J,I
8290 PQ%=2:KK=3:GOSUB 1590
8300 PQ%=0:IF EE%>0 THEN 600
8310 PQ%=2:PP%=4:KK=5:GOSUB 640
8320 PQ%=0:MM=1:GOSUB 4590
8330 MM=2:GOSUB 4590
8340 MM=8:GOSUB 4710
8350 FOR I=1 TO N:B(I,I)=1!:NEXT I:TT=0:KT=0
8360 FOR I=1 TO N-1
8370 FOR J=I+1 TO N:TT=TT+A(I,J)^2:NEXT J,I:TT=SQR(TT)*2:QQ=.0001*TT/N
8380 TT=TT/N
8390 FOR I=1 TO N-1
8400 FOR J=I+1 TO N:X=A(I,J):IF ABS(X)<=TT THEN 8460 ELSE X=-X:Y=.5*(A(I,I)-A(J,
J)):T=SGN(Y)*X/SQR(X*X+Y*Y):X=T/SQR(2*(1+SQR(1-T*T))
):Y=SQR(1-X*X)
8410 FOR II=1 TO N:T=B(II,I):Q=B(II,J):B(II,I)=T*Y-Q*X:B(II,J)=T*X+Q*Y:T=A(II,I)
:Q=A(II,J):A(II,I)=T*Y-Q*X:A(II,J)=T*X+Q*Y:NEXT II
8420 FOR II=1 TO N:T=A(I,II):Q=A(J,II)
8430 A(I,II)=Y*T-X*Q:A(J,II)=X*T+Y*Q
8450 NEXT II:KT=KT+1
8460 NEXT J,I:IF KT=0 THEN 8470 ELSE KT=0:GOTO 8360
8470 IF TT>QQ THEN 8380 ELSE 7490
8480 PRINT CHR$(140):RETURN
8520 PRINT:PRINT "*To CONTINUE":GOTO 7500
8530 PRINT:PRINT "**** OUT OF MEMORY.";:R$="Pls. FILE or ERASE Unneeded Matrices
"
8540 TT%(0,MM)=0:TT%(1,MM)=0:IF MM<>0 AND MM<>8 THEN 8550 ELSE NR=NR+1:NC=NC+1
8550 AA%=AA%-NR*NC:RESUME 820
8560 IF PQ%>0 THEN RETURN ELSE 210
10000 PRINT:PRINT"*USER MODULE is NON-EXISTENT.":PQ%=0:GOTO 7500
```