ALGOL MATRIX PACKAGE(ALMAT).

Elliott provided ALGOL users with a source code library of procedures for matrix calculations. This library, called ALMAT originated with the Elliott 803 and 503 computers that preceded the 900 series.

The contents of the library are as follows:

```
procedure mxcopy (A) becomes: (B)
array A, B;
```

Copy the contents of matrix B to matrix A. A, and B must both be 2 dimensional arrays with the same ranges in each of their dimensions.

```
procedure mxneg (A) becomes minus: (B);
array A, B;
```

Calculate the value of –B and store in A. A, and B must both be 2 dimensional arrays with the same ranges in each of their dimensions.

```
procedure mxsum (A) becomes: (B) minus: (C);
array A, B, C;
```

Calculate the value of A + B and store the result in A. A, B and C must both be 2 dimensional arrays with the same ranges in each of their dimensions.

```
procedure mxdiff (A) becomes: (B) minus: (C);
array A, B, C;
```

Calculate the result of B - C and store the result in A. A, B and C must both be 2 dimensional arrays with the same ranges in each of their dimensions.

```
procedure mxprod (A) becomes: (B) times: (C);
array A, B, C;
```

Calculate the result of B * C and store the result in A. A, B and C must all be 2 dimensional arrays with the same ranges in each of their dimensions. A must be distinct from B or C.

```
procedure mxtrans (A) becomes transpose of: (B);
array A, B;
```

Store the transpose of matrix B in A. A, and B must both be 2 dimensional arrays with the same ranges in each of their dimensions. B must be distinct from A.

```
procedure scprod (A) becomes: (B) times the scalar: (X);
value X; real X; array A;
```

Calculate the result of X * B the result in A. A must be a 2 dimensional array.

```
procedure invmx (A);
array A;
```

Invert the matrix A in situ. A must be square, i.e., a 2 dimensional array with the same range for each dimension.

```
procedure solvmx (A, B);
array A;
```

Solve the equation AX=B and store the result in B. (A is also overwritten). A and B must be distinct 2 dimensional arrays with the same ranges for their first dimensions. A must be square, i.e., the same ranges for both its dimensions. B must be a vector, i.e., have range 1 for its second dimension, as in array B [1:n, 1:1].

```
procedure mxquot (B, A, C);
array A;
```

Solve the equation AB=C and store the result in B. (A is overwritten, C is preserved). A, B and C must both be 2 dimensional arrays with the same ranges in the first dimension. B and C must be vectors, i.e., have range 1 in their second dimension, as in array B [1:n, 1:1].

```
procedure readmx (A);
array A;
```

Input the matrix A from current reader. A must be a 2 dimensional array.

```
procedure printmx (A);
array A;
```

Output the matrix A to the current punch in row order. A must be a 2 dimensional array. Each row of the matrix is output on a separate line. The user should set an appropriate format for printing each element of the matrix.

```
procedure printcol (A);
array A;
```

Output the matrix A to the current punch in column order. A must be a 2 dimensional array. A must be a 2 dimensional array. Each column of the matrix is output on a separate line. The format for printing each element should be set by the programmer.

```
procedure spaces (T);
value T; integer T;
```

Output T spaces to the current punch.

```
procedure mxoutput (A, M, N);
array A; value M, N; integer M, N;
```

Output the matrix A to the current punch. A must be a 2 dimensional array. M should be set to the number of characters available across the output sheet and N the number of characters occupied by each element. The format for printing each element should be set by the programmer. From this the number of characters required for each element can be calculated, e.g.,

aligned(R, S)	R+S+2 characters
freepoint(T)	T+2 characters
scaled(T)	T+6 characters.

The matrix is output in row order with column and row numbers included preceded by 10 blank lines and followed by 4 blank lines, the text "END OF MATRIX" and 2 further blank lines. The rows are double spaced.

In addition to these procedures, ALMAT contains a number of internal variables and procedures which the programmer should avoid using: det, innsum, error, print1, print2, permrows, crout and spaces.

ALMAT is a source code library consisting of a partial program terminated by a halt code. It can be found in the file ALMAT.903 in the directory ALMAT. The programmer should initially load ALMAT for translation. When the ALMAT tape has been read, the programmer should then read in his own program which should be followed by 2 * end statements to close the blocks containing ALMAT.

DEMO5.DAT in the 903ALGOL directory illustrates the use of ALMAT using large program mode to allow for large matrices (80 * 80).