1 SUMMARY

Appends an n+1 **vector to an** $n\times n$ **triangular matrix** to form an (n+1)×(n+1) triangular matrix, i.e. given an upper triangular matrix $\mathbf{U}=\{u_{ij}\}_{n\times n}$ and a vector $\mu=\mu_1,\mu_2,...,\mu_{n+1}$ this subroutine forms the triangular matrix

$$\tilde{\mathbf{U}} = \left(\begin{array}{c|c} \mathbf{U} & \\ \hline 0 & \mu \end{array}\right)$$

Both $\tilde{\mathbf{U}}$ and $\mathbf{U}$ are stored in compact form.

ATTRIBUTES — **Version:** 1.0.0. **Remark:** the matrix storage format is that of `MC11`. **Types:** `MC16A, MC16AD`. **Original date:** September 1974. **Origin:** M.J.D.Powell, Harwell.

2 HOW TO USE THE PACKAGE

2.1 The argument list

The single precision version

```
CALL MC16A(A,N,COL)
```

The double precision version

```
CALL MC16AD(A,N,COL)
```

`A` is a `REAL` (`DOUBLE PRECISION` in the D version) one dimensional array of length at least (n+1)(n+2)/2, whose first n(n+1)/2 elements represent $\mathbf{U}$. It is convenient to use the notation d_1, $l_{2,1}$, $l_{3,1}$,..., $l_{n,1}$, d_2, $l_{3,2}$,..., $l_{n,2}$,..., d_n for these elements, to let $\mathbf{D}$ be the diagonal matrix whose diagonal elements are d_1, d_2,..., d_n, and to let $\mathbf{L}$ be the lower triangular matrix with ones on the diagonal whose other non-zero elements are l_{ij} (i>j). Then $\mathbf{D}$ and $\mathbf{L}$ are related to $\mathbf{U}$ by the equation $\mathbf{U}^T\mathbf{U}=\mathbf{L}\mathbf{D}\mathbf{L}^T$. On exit from the subroutine the first (n+1)(n+2)/2 elements of `A` represent the required upper triangular matrix, in the $\mathbf{D}$–$\mathbf{L}$ form that is used for input.

`N` is an `INTEGER` that is initially set to the dimension of $\mathbf{U}$. The subroutine increases its value by one to correspond to the dimension of the new matrix. **Restriction:** n>0.

`COL` is `REAL` (`DOUBLE PRECISION` in the D version) one dimensional array, whose first (n+1) elements are the components of the vector that is appended to $\mathbf{U}$. It is unchanged by the subroutine.

3 GENERAL INFORMATION

Workspace: The total amount of work is a small multiple of n^2, due to shifting the information in the array `A`.

Use of common: None.

Other routines called directly: None.

Input/output: Other.

Restrictions: n>0. There is no upper bound on the value of n.