

M01EAF – NAG Fortran Library Routine Document

Note. Before using this routine, please read the Users' Note for your implementation to check the interpretation of bold italicised terms and other implementation-dependent details.

1 Purpose

M01EAF rearranges a vector of *real* numbers into the order specified by a vector of ranks.

2 Specification

```
SUBROUTINE M01EAF(RV, M1, M2, IRANK, IFAIL)
  INTEGER          M1, M2, IRANK(M2), IFAIL
  real           RV(M2)
```

3 Description

M01EAF is designed to be used typically in conjunction with the M01D- ranking routines. After one of the M01D- routines has been called to determine a vector of ranks, M01EAF can be called to rearrange a vector of real numbers into the rank order. If the vector of ranks has been generated in some other way, then M01ZBF should be called to check its validity before M01EAF is called.

4 References

None.

5 Parameters

- 1: RV(M2) — *real* array *Input/Output*
On entry: elements M1 to M2 of RV must contain *real* values to be rearranged.
On exit: these values are rearranged into rank order. For example, if $IRANK(i) = M1$, then the initial value of $RV(i)$ is moved to $RV(M1)$.
- 2: M1 — INTEGER *Input*
- 3: M2 — INTEGER *Input*
On entry: M1 and M2 must specify the range of the ranks supplied in IRANK and the elements of RV to be rearranged.
Constraint: $0 < M1 \leq M2$.
- 4: IRANK(M2) — INTEGER array *Input*
On entry: elements M1 to M2 of IRANK must contain a permutation of the integers M1 to M2, which are interpreted as a vector of ranks.
- 5: IFAIL — INTEGER *Input/Output*
On entry: IFAIL must be set to 0, -1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.
On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

If on entry `IFAIL = 0` or `-1`, explanatory error messages are output on the current error message unit (as defined by `X04AAF`).

Errors detected by the routine:

`IFAIL = 1`

On entry, `M2 < 1`,
or `M1 < 1`,
or `M1 > M2`.

`IFAIL = 2`

Elements `M1` to `M2` of `IRANK` contain a value outside the range `M1` to `M2`.

`IFAIL = 3`

Elements `M1` to `M2` of `IRANK` contain a repeated value.

If `IFAIL = 2` or `3`, elements `M1` to `M2` of `IRANK` do not contain a permutation of the integers `M1` to `M2`. On exit, the contents of `RV` may be corrupted. To check the validity of `IRANK` without the risk of corrupting `RV`, use `M01ZBF`.

7 Accuracy

Not applicable.

8 Further Comments

The average time taken by the routine is approximately proportional to n , where $n = M2 - M1 + 1$.

9 Example

The example program reads a matrix of *real* numbers and rearranges its rows so that the elements of the k th column are in ascending order. To do this, the program first calls `M01DAF` to rank the elements of the k th column, and then calls `M01EAF` to rearrange each column into the order specified by the ranks. The value of k is read from the data-file.

9.1 Program Text

Note. The listing of the example program presented below uses bold italicised terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*      M01EAF Example Program Text
*      Mark 14 Revised.  NAG Copyright 1989.
*      .. Parameters ..
      INTEGER          MMAX, NMAX
      PARAMETER       (MMAX=20,NMAX=20)
      INTEGER          NIN, NOUT
      PARAMETER       (NIN=5,NOUT=6)
*      .. Local Scalars ..
      INTEGER          I, IFAIL, J, K, M, N
*      .. Local Arrays ..
      real             RM(MMAX,NMAX)
      INTEGER          IRANK(MMAX)
*      .. External Subroutines ..
      EXTERNAL         M01DAF, M01EAF
*      .. Executable Statements ..
```

```

        WRITE (NOUT,*) 'M01EAF Example Program Results'
*      Skip heading in data file
        READ (NIN,*)
        READ (NIN,*) M, N, K
        IF (M.GE.1 .AND. M.LE.MMAX .AND. N.GE.1 .AND. N.LE.NMAX .AND.
+       K.GE.1 .AND. K.LE.N) THEN
          DO 20 I = 1, M
            READ (NIN,*) (RM(I,J),J=1,N)
20      CONTINUE
          IFAIL = 0
*
          CALL M01DAF(RM(1,K),1,M,'Ascending',IRANK,IFAIL)
*
          DO 40 J = 1, N
*
            CALL M01EAF(RM(1,J),1,M,IRANK,IFAIL)
*
40      CONTINUE
        WRITE (NOUT,*)
        WRITE (NOUT,99999) 'Matrix sorted on column', K
        WRITE (NOUT,*)
        DO 60 I = 1, M
          WRITE (NOUT,99998) (RM(I,J),J=1,N)
60      CONTINUE
        END IF
        STOP
*
99999 FORMAT (1X,A,I3)
99998 FORMAT (1X,3F7.1)
        END

```

9.2 Program Data

M01EAF Example Program Data

```

12 3 1
6.0 5.0 4.0
5.0 2.0 1.0
2.0 4.0 9.0
4.0 9.0 6.0
4.0 9.0 5.0
4.0 1.0 2.0
3.0 4.0 1.0
2.0 4.0 6.0
1.0 6.0 4.0
9.0 3.0 2.0
6.0 2.0 5.0
4.0 9.0 6.0

```

9.3 Program Results

M01EAF Example Program Results

Matrix sorted on column 1

```

1.0    6.0    4.0
2.0    4.0    9.0
2.0    4.0    6.0

```

3.0	4.0	1.0
4.0	9.0	6.0
4.0	9.0	5.0
4.0	1.0	2.0
4.0	9.0	6.0
5.0	2.0	1.0
6.0	5.0	4.0
6.0	2.0	5.0
9.0	3.0	2.0
