M01EBF – 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

M01EBF rearranges a vector of integer numbers into the order specified by a vector of ranks.

2 Specification

```
SUBROUTINE M01EBF(IV, M1, M2, IRANK, IFAIL)
INTEGER           IV(M2), M1, M2, IRANK(M2), IFAIL
```

3 Description

M01EBF 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, M01EBF can be called to rearrange a vector of integer 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 M01EBF is called.

4 References

None.

5 Parameters

1: IV(M2) — INTEGER array *Input/Output*

On entry: elements M1 to M2 of IV must contain integer values to be rearranged.

On exit: these values are rearranged into rank order. For example, if IRANK(i) = M1, then the initial value of IV(i) is moved to IV(M1).

2: M1 — INTEGER *Input*

3: M2 — INTEGER *Input*

On entry: M1 and M2 specify the range of the ranks supplied in IRANK and the elements of IV to be rearranged.

Constraint: $0 < \text{M1} \leq \text{M2}$.

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 IV may be corrupted. To check the validity of IRANK without the risk of corrupting IV, use M01ZBF.

7 Accuracy

Not applicable.

8 Further Comments

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

9 Example

The example program reads a matrix of integers and rearranges its rows so that the elements of the kth column are in ascending order. To do this, the program first calls M01DBF to rank the elements of the kth column, and then calls M01EBF 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.

```
*     M01EBF 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 ..
      INTEGER          IM(MMAX,NMAX), IRANK(MMAX)
*     .. External Subroutines ..
      EXTERNAL         M01DBF, M01EBF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'M01EBF 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,*) (IM(I,J),J=1,N)
   20    CONTINUE
         IFAIL = 0
*
         CALL M01DBF(IM(1,K),1,M,'Ascending',IRANK,IFAIL)
*
         DO 40 J = 1, N
*
            CALL M01EBF(IM(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) (IM(I,J),J=1,N)
   60    CONTINUE
      END IF
      STOP
*
99999 FORMAT (1X,A,I3)
99998 FORMAT (1X,3I7)
      END
```

9.2 Program Data

```
M01EBF Example Program Data
12 3 1
6 5 4
5 2 1
2 4 9
4 9 6
4 9 5
4 1 2
3 4 1
2 4 6
1 6 4
9 3 2
6 2 5
4 9 6
```

9.3 Program Results

```
 M01EBF Example Program Results

 Matrix sorted on column  1

       1      6      4
       2      4      9
       2      4      6
       3      4      1
```

```
4       9       6
4       9       5
4       1       2
4       9       6
5       2       1
6       5       4
6       2       5
9       3       2
```
