M01ZBF – 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

M01ZBF checks the validity of a permutation.

2   Specification

```
SUBROUTINE M01ZBF(IPERM, M1, M2, IFAIL)
INTEGER          IPERM(M2), M1, M2, IFAIL
```

3   Description

M01ZBF can be used to check the validity of user-supplied ranks or indices, without the ranks or indices being corrupted.

4   References

None.

5   Parameters

1:   IPERM(M2) — INTEGER array   *Input*

On entry: elements M1 to M2 of IPERM must be set to values which are supposed to be a permutation of the integers M1 to M2. If they are a valid permutation, the routine exits with IFAIL = 0.

2:   M1 — INTEGER   *Input*
3:   M2 — INTEGER   *Input*

On entry: the range of elements used in the array IPERM and the range of values in the permutation, as specified under IPERM.

Constraint: $0 < M1 \leq M2$.

4:   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 IPERM contain a value outside the range M1 to M2.

IFAIL = 3

   Elements M1 to M2 of IPERM contain a repeated value.

If IFAIL = 2 or 3, elements M1 to M2 of IPERM do not contain a permutation of the integers M1 to M2.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

The example program reads in a vector of **real** numbers, and a vector of ranks; it calls M01ZBF to check the validity of the ranks before calling M01EAF to rearrange the **real** numbers into the specified order.

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.

```
*      M01ZBF Example Program Text
*      Mark 14 Revised.  NAG Copyright 1989.
*      .. Parameters ..
       INTEGER          NMAX
       PARAMETER        (NMAX=100)
       INTEGER          NIN, NOUT
       PARAMETER        (NIN=5,NOUT=6)
*      .. Local Scalars ..
       INTEGER          I, IFAIL, N
*      .. Local Arrays ..
       real             RV(NMAX)
       INTEGER          IRANK(NMAX)
*      .. External Subroutines ..
       EXTERNAL         M01EAF, M01ZBF
*      .. Executable Statements ..
       WRITE (NOUT,*) 'M01ZBF Example Program Results'
*      Skip heading in data file
       READ (NIN,*)
       READ (NIN,*) N
       IF (N.GE.1 .AND. N.LE.NMAX) THEN
          READ (NIN,*) (RV(I),I=1,N)
          READ (NIN,*) (IRANK(I),I=1,N)
          IFAIL = 0
*
          CALL M01ZBF(IRANK,1,N,IFAIL)
          CALL M01EAF(RV,1,N,IRANK,IFAIL)
*
          WRITE (NOUT,*)
          WRITE (NOUT,*) 'Numbers in rank order'
          WRITE (NOUT,*)
```

```
      WRITE (NOUT,99999) (RV(I),I=1,N)
    END IF
    STOP
*
99999 FORMAT (1X,10F7.1)
    END
```

9.2   Program Data

```
M01ZBF Example Program Data
12
5.3 4.6 7.8 1.7 5.3 9.9 3.2 4.3 7.8 4.5 1.2 7.6
  7   6  10   2   8  12   3   4  11   5   1   9
```

9.3   Program Results

```
M01ZBF Example Program Results

Numbers in rank order

    1.2    1.7    3.2    4.3    4.5    4.6    5.3    5.3    7.6    7.8
    7.8    9.9
```