NAG Fortran Library Routine Document

M01DBF

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

M01DBF ranks a vector of integer numbers in ascending or descending order.

2 Specification

```
SUBROUTINE M01DBF(IV, M1, M2, ORDER, IRANK, IFAIL)
INTEGER           IV(M2), M1, M2, IRANK(M2), IFAIL
CHARACTER*1       ORDER
```

3 Description

M01DBF uses a variant of list-merging, as described by Knuth (1973) pp 165-166. The routine takes advantage of natural ordering in the data, and uses a simple list insertion in a preparatory pass to generate ordered lists of length at least 10. The ranking is stable: equal elements preserve their ordering in the input data.

4 References

Knuth D E (1973) *The Art of Computer Programming (Volume 3)* (2nd Edition) Addison-Wesley

5 Parameters

1: IV(M2) – INTEGER array *Input*

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

2: M1 – INTEGER *Input*

On entry: the index of the first element of IV to be ranked.

Constraint: M1 > 0.

3: M2 – INTEGER *Input*

On entry: M2 must specify the index of the last element of IV to be ranked.

Constraint: M2 $\geq$ M1.

4: ORDER – CHARACTER*1 *Input*

On entry: if ORDER is 'A', the values will be ranked in ascending (i.e., non-decreasing) order; if ORDER is 'D', into descending order.

Constraint: ORDER $=$ 'A' or 'D'.

5: IRANK(M2) – INTEGER array *Output*

On exit: elements M1 to M2 of IRANK contain the ranks of the corresponding elements of IV. Note that the ranks are in the range M1 to M2: thus, if IV(i) is the first element in the rank order, IRANK(i) is set to M1.

6: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

For environments where it might be inappropriate to halt program execution when an error is detected, the value −1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, for users not familiar with this parameter the recommended value is 0. **When the value −1 or 1 is used it is essential to test the value of IFAIL on exit.**

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 or warnings detected by the routine:

IFAIL = 1

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

IFAIL = 2

On entry, ORDER is not 'A' or 'D'.

7 Accuracy

Not applicable.

8 Further Comments

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

9 Example

The example program reads a list of integers and ranks them in descending 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.

```
*     M01DBF 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 ..
      INTEGER          IRANK(NMAX), IV(NMAX)
*     .. External Subroutines ..
      EXTERNAL         M01DBF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'M01DBF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
```

```
      READ (NIN,*) N
      IF (N.GE.1 .AND. N.LE.NMAX) THEN
         READ (NIN,*) (IV(I),I=1,N)
         IFAIL = 0
*
         CALL M01DBF(IV,1,N,'Descending',IRANK,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) '   Data   Ranks'
         WRITE (NOUT,*)
         DO 20 I = 1, N
            WRITE (NOUT,99999) IV(I), IRANK(I)
   20    CONTINUE
      END IF
      STOP
*
99999 FORMAT (1X,2I7)
      END
```

9.2 Program Data

```
M01DBF Example Program Data
12
34 44 89 64 69 69 23 1 999 65 22 76
```

9.3 Program Results

```
 M01DBF Example Program Results

    Data   Ranks

      34      9
      44      8
      89      2
      64      7
      69      4
      69      5
      23     10
       1     12
     999      1
      65      6
      22     11
      76      3
```