NAG Fortran Library Routine Document

M01CAF

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

M01CAF rearranges a vector of ***real*** numbers into ascending or descending order.

2 Specification

```
SUBROUTINE M01CAF(RV, M1, M2, ORDER, IFAIL)
INTEGER             M1, M2, IFAIL
real                RV(M2)
CHARACTER*1         ORDER
```

3 Description

M01CAF is based on Singleton's implementation of the 'median-of-three' Quicksort algorithm (Singleton (1969)), but with two additional modifications. First, small subfiles are sorted by an insertion sort on a separate final pass (Sedgewick (1978)). Second, if a subfile is partitioned into two very unbalanced subfiles, the larger of them is flagged for special treatment: before it is partitioned, its end-points are swapped with two random points within it; this makes the worst case behaviour extremely unlikely.

4 References

Sedgewick R (1978) Implementing Quicksort programs *Comm. ACM* **21** 847–857

Singleton R C (1969) An efficient algorithm for sorting with minimal storage: Algorithm 347 *Comm. ACM* **12** 185–187

5 Parameters

1: RV(M2) – ***real*** array *Input/Output*

On entry: elements M1 to M2 of RV must contain ***real*** values to be sorted.

On exit: these values are rearranged into sorted order.

2: M1 – INTEGER *Input*

On entry: the index of the first element of RV to be sorted.

Constraint: $\mathrm{M1} > 0$.

3: M2 – INTEGER *Input*

On entry: the index of the last element of RV to be sorted.

Constraint: $\mathrm{M2} \geq \mathrm{M1}$.

4: ORDER – CHARACTER*1 *Input*

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

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

5: 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 $n =$ M2 $-$ M1 $+ 1$. The worst case time is proportional to n^2 but this is extremely unlikely to occur.

9 Example

The example program reads a list of ***real*** numbers and sorts them into ascending 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.

```
*     M01CAF 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)
*     .. External Subroutines ..
      EXTERNAL         M01CAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'M01CAF 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)
         IFAIL = 0
*
         CALL M01CAF(RV,1,N,'Ascending',IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Sorted numbers'
         WRITE (NOUT,*)
         WRITE (NOUT,99999) (RV(I),I=1,N)
      END IF
      STOP
*
99999 FORMAT (1X,10F7.1)
      END
```

9.2 Program Data

```
M01CAF Example Program Data
16
1.3 5.9 4.1 2.3 0.5 5.8 1.3 6.5
2.3 0.5 6.5 9.9 2.1 1.1 1.2 8.6
```

9.3 Program Results

```
 M01CAF Example Program Results

 Sorted numbers

     0.5    0.5    1.1    1.2    1.3    1.3    2.1    2.3    2.3    4.1
     5.8    5.9    6.5    6.5    8.6    9.9
```