NAG 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 (KIND=nag_wp) RV(M2)
CHARACTER(1)        ORDER
```

3 Description

M01CAF is based on Singleton's implementation of the 'median-of-three' Quicksort algorithm (see Singleton (1969)), but with two additional modifications. First, small subfiles are sorted by an insertion sort on a separate final pass (see 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 (KIND=nag_wp) 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: M1 > 0.

3: M2 – INTEGER *Input*

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

Constraint: M2 $\geq$ M1.

4: ORDER – CHARACTER(1) *Input*

On entry: if ORDER = 'A', the values will be sorted into ascending (i.e., nondecreasing) order.

If ORDER = '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. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

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, if you are 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.**

On exit: IFAIL = 0 unless the routine detects an error or a warning has been flagged (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 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 M01CAF is approximately proportional to $n \times \log n$, where $n = \text{M2} - \text{M1} + 1$. The worst case time is proportional to n^2 but this is extremely unlikely to occur.

9 Example

This example reads a list of real numbers and sorts them into ascending order.

9.1 Program Text

```
    Program m01cafe

!     M01CAF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: m01caf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                :: nin = 5, nout = 6
!     .. Local Scalars ..
```

```
      Integer                          :: i, ifail, m1, m2
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: rv(:)
!     .. Executable Statements ..
      Write (nout,*) 'M01CAF Example Program Results'

!     Skip heading in data file
      Read (nin,*)

      Read (nin,*) m2
      Allocate (rv(m2))

      m1 = 1

      Read (nin,*)(rv(i),i=m1,m2)

      ifail = 0
      Call m01caf(rv,m1,m2,'Ascending',ifail)

      Write (nout,*)
      Write (nout,*) 'Sorted numbers'
      Write (nout,*)
      Write (nout,99999)(rv(i),i=m1,m2)

99999 Format (1X,10F7.1)
    End Program m01cafe
```

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
```