

NAG Toolbox for MATLAB

m01cb

1 Purpose

m01cb rearranges a vector of integer numbers into ascending or descending order.

2 Syntax

```
[iv, ifail] = m01cb(iv, m1, order, 'm2', m2)
```

3 Description

m01cb 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

5.1 Compulsory Input Parameters

- 1: **iv(m2)** – **int32** array
Elements **m1** to **m2** of **iv** must contain integer values to be sorted.
- 2: **m1** – **int32** scalar
The index of the first element of **iv** to be sorted.
Constraint: **m1** > 0.
- 3: **order** – **string**
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'.

5.2 Optional Input Parameters

- 1: **m2** – **int32** scalar
Default: The dimension of the array **iv**.
the index of the last element of **iv** to be sorted.
Constraint: **m2** ≥ **m1**.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **iv(m2)** – **int32** array

These values are rearranged into sorted order.

2: **ifail** – **int32** scalar

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

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 function is approximately proportional to $n \times \log n$, where $n = \mathbf{m2} - \mathbf{m1} + 1$. The worst case time is proportional to n^2 but this is extremely unlikely to occur.

9 Example

```
iv = [int32(23);
      int32(45);
      int32(45);
      int32(67);
      int32(69);
      int32(90);
      int32(999);
      int32(1);
      int32(78);
      int32(112);
      int32(24);
      int32(69);
      int32(96);
      int32(99);
      int32(45);
      int32(78)];
m1 = int32(1);
order = 'Descending';
[ivOut, ifail] = m01cb(iv, m1, order)
```

```
ivOut =
      999
      112
       99
       96
       90
       78
```

	78
	69
	69
	67
	45
	45
	45
	24
	23
	1
ifail =	0