

NAG Toolbox for MATLAB

m01za

1 Purpose

m01za inverts a permutation, and hence converts a rank vector to an index vector, or vice versa.

2 Syntax

```
[iperm, ifail] = m01za(iperm, m1, 'm2', m2)
```

3 Description

There are two common ways of describing a permutation using an integer vector **iperm**. The first uses ranks: **iperm**(*i*) holds the position to which the *i*th data element should be moved in order to sort the data; in other words its rank in the sorted order. The second uses indices: **iperm**(*i*) holds the current position of the data element which would occur in *i*th position in sorted order. For example, given the values

3.5 5.9 2.9 0.5

to be sorted in ascending order, the ranks would be

3 4 2 1

and the indices would be

4 3 1 2

The M01D functions generate ranks, and the M01E functions require ranks to be supplied to specify the reordering. However if it is desired simply to refer to the data in sorted order without actually reordering them, indices are more convenient than ranks (see the example in Section 9).

m01za can be used to convert ranks to indices, or indices to ranks, as the two permutations are inverses of each another.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

- 1: **iperm(m2)** – int32 array

Elements **m1** to **m2** of **iperm** must contain a permutation of the integers **m1** to **m2**.

- 2: **m1** – int32 scalar

m1 and **m2** must specify the range of elements used in the array **iperm** and the range of values in the permutation, as specified under **iperm**.

Constraint: $0 < \mathbf{m1} \leq \mathbf{m2}$.

5.2 Optional Input Parameters

- 1: **m2** – int32 scalar

Default: The dimension of the array **iperm**.

m1 and **m2** must specify the range of elements used in the array **iperm** and the range of values in the permutation, as specified under **iperm**.

Constraint: $0 < \mathbf{m1} \leq \mathbf{m2}$.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

- 1: **iperm(m2)** – int32 array

These elements contain the inverse permutation of the integers **m1** to **m2**.

- 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

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**; on exit these elements are usually corrupted. To check the validity of a permutation without the risk of corrupting it, use m01zb.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

```
iperm = [int32(11);  
         int32(9);  
         int32(3);  
         int32(7);  
         int32(6);  
         int32(5);  
         int32(4);  
         int32(2);  
         int32(1);  
         int32(12);
```

```
        int32(10);  
        int32(8)];  
m1 = int32(1);  
[ipermOut, ifail] = m01za(iperm, m1)
```

```
ipermOut =  
          9  
          8  
          3  
          7  
          6  
          5  
          4  
         12  
          2  
         11  
          1  
         10  
  
ifail =  
        0
```