NAG Toolbox for MATLAB

m01zb

1 Purpose

m01zb checks the validity of a permutation.

2 Syntax

```
[iperm, ifail] = m01zb(iperm, m1, 'm2', m2)
```

3 Description

m01zb can be used to check the validity of user-supplied ranks or indices, without the ranks or indices being corrupted.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **iperm**(**m2**) – **int32 array**

Elements **m1** to **m2** of **iperm** must be set to values which are supposed to be a permutation of the integers **m1** to **m2**. If they are a valid permutation, the function exits with **ifail** $= 0$.

2: **m1** – **int32 scalar**

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

Constraint: $0 <$ **m1** $\leq$ **m2**.

5.2 Optional Input Parameters

1: **m2** – **int32 scalar**

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

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

Constraint: $0 <$ **m1** $\leq$ **m2**.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **iperm**(**m2**) – **int32 array**

Used as internal workpsace prior to being restored and hence is unchanged.

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**.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

```
iperm = [int32(7);
     int32(6);
     int32(10);
     int32(2);
     int32(8);
     int32(12);
     int32(3);
     int32(4);
     int32(11);
     int32(5);
     int32(1);
     int32(9)];
m1 = int32(1);
[ipermOut, ifail] = m01zb(iperm, m1)
```

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