NAG Library Routine Document

M01NBF

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

M01NBF searches an ordered vector of integer numbers and returns the index of the first value equal to the sought-after item.

2 Specification

```
FUNCTION M01NBF (VALID, IV, M1, M2, ITEM, IFAIL)
INTEGER M01NBF
INTEGER IV(M2), M1, M2, ITEM, IFAIL
LOGICAL VALID
```

3 Description

M01NBF is based on Professor Niklaus Wirth's implementation of the Binary Search algorithm (see Wirth (2004)), but with two modifications. First, if the sought-after item is less than the value of the first element of the array to be searched, 0 is returned. Second, if a value equal to the sought-after item is not found, the index of the immediate lower value is returned.

4 References

Wirth N (2004) *Algorithms and Data Structures* 35–36 Prentice Hall

5 Parameters

1: VALID – LOGICAL *Input*

On entry: if VALID is set to .TRUE. parameter checking will be performed. If VALID is set to .FALSE. M01NBF will be called without parameter checking (which includes checking that array IV is sorted in ascending order) and the routine will return with IFAIL $= 0$. See Section 9 for further details.

2: IV(M2) – INTEGER array *Input*

On entry: elements M1 to M2 contain integer values to be searched.

Constraint: elements M1 to M2 of IV must be sorted in ascending order.

3: M1 – INTEGER *Input*

On entry: the index of the first element of IV to be searched.

Constraint: M1 ≥ 1.

4: M2 – INTEGER *Input*

On entry: the index of the last element of IV to be searched.

Constraint: M2 $\geq$ M1.

5: ITEM – INTEGER *Input*

On entry: the sought-after item.

6: 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:

(**Note:** these errors will only be returned if VALID = .TRUE..)

IFAIL $= 2$

On entry, IV must be sorted in ascending order: IV element $\langle value \rangle >$ element $\langle value \rangle$.

IFAIL $= 3$

On entry, M1 $= \langle value \rangle$.
Constraint: M1 ≥ 1.

IFAIL $= 4$

On entry, M1 $= \langle value \rangle$, M2 $= \langle value \rangle$.
Constraint: M1 $\leq$ M2.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

Not applicable.

8 Parallelism and Performance

Not applicable.

9 Further Comments

The argument VALID should be used with caution. Set it to .FALSE. only if you are confident that the other arguments are correct, in particular that array IV is in fact arranged in ascending order. If you wish to search the same array IV many times, you are recommended to set VALID to .TRUE. on first call of M01NBF and to .FALSE. on subsequent calls, in order to minimize the amount of time spent checking IV, which may be significant if IV is large.

The time taken by M01NBF is $O(\log(n))$, where $n = \mathrm{M2} - \mathrm{M1} + 1$, when VALID = .FALSE..

10 Example

This example reads a list of integer numbers and sought-after items and performs the search for these items.

10.1 Program Text

```
    Program m01nbfe

!     M01NBF Example Program Text
!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: m01nbf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, index, ioerr, item, m1, m2
      Logical                          :: first
!     .. Local Arrays ..
      Integer, Allocatable             :: iv(:)
!     .. Executable Statements ..
      Write (nout,*) 'M01NBF Example Program Results'

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

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

      m1 = 1

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

      first = .True.

data: Do
        Read (nin,*,Iostat=ioerr) item

        If (ioerr<0) Then
          Exit data
        End If

        ifail = 0
        index = m01nbf(first,iv,m1,m2,item,ifail)

        If (first) Then
          Write (nout,*)
          Write (nout,*) 'Reference Vector is:'
          Write (nout,99999)(iv(i),i=m1,m2)
          first = .False.
        End If

        Write (nout,*)
        Write (nout,99998) item, index
```

```
      End Do data

99999 Format (1X,8I5)
99998 Format (1X,'Search for item ',I5,' returned index: ',I4)
    End Program m01nbfe
```

10.2 Program Data

```
M01NBF Example Program Data
16                                                  : M2
5 6 11 12 13 13 21 23
23 41 58 59 65 65 86 99                             : IV
21                                                  : Item 1
4                                                   : Item 2
71                                                  : Item 3
100                                                 : Item 4
```

10.3 Program Results

```
 M01NBF Example Program Results

 Reference Vector is:
     5    6   11   12   13   13   21   23
    23   41   58   59   65   65   86   99

 Search for item    21 returned index:    7

 Search for item     4 returned index:    0

 Search for item    71 returned index:   14

 Search for item   100 returned index:   16
```