NAG Library Routine Document

F16DNF

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

F16DNF computes the largest component of an integer vector, along with the index of that component.

2 Specification

```
SUBROUTINE F16DNF (N, X, INCX, K, I)
INTEGER N, X(1+(N-1)*ABS(INCX)), INCX, K, I
```

3 Description

F16DNF computes the largest component, i, of an n-element integer vector x, and determines the smallest index, k, such that

$$i = x_k = \max_j x_j.$$

4 References

Basic Linear Algebra Subprograms Technical (BLAST) Forum (2001) *Basic Linear Algebra Subprograms Technical (BLAST) Forum Standard* University of Tennessee, Knoxville, Tennessee http://www.netlib.org/blas/blast-forum/blas-report.pdf

5 Arguments

1: N – INTEGER *Input*

On entry: n, the number of elements in x.

2: X$(1 + (\text{N} - 1) \times |\text{INCX}|)$ – INTEGER array *Input*

On entry: the n-element vector x.

If INCX > 0, x_i must be stored in X$((i-1) \times \text{INCX} + 1)$, for $i = 1, 2, \ldots, \text{N}$.

If INCX < 0, x_i must be stored in X$((\text{N} - i) \times |\text{INCX}| + 1)$, for $i = 1, 2, \ldots, \text{N}$.

Intermediate elements of X are not referenced. If N $= 0$, X is not referenced.

3: INCX – INTEGER *Input*

On entry: the increment in the subscripts of X between successive elements of x.

Constraint: INCX $\neq 0$.

4: K – INTEGER *Output*

On exit: k, the index, from the set $\{1, 2, \ldots, \text{N}\}$, of the largest component of x. If N ≤ 0 on input then K is returned as 0.

5: I – INTEGER *Output*

On exit: i, the largest component of x. If N ≤ 0 on input then I is returned as 0.

6 Error Indicators and Warnings

If INCX $= 0$, an error message is printed and program execution is terminated.

7 Accuracy

The BLAS standard requires accurate implementations which avoid unnecessary over/underflow (see Section 2.7 of Basic Linear Algebra Subprograms Technical (BLAST) Forum (2001)).

8 Parallelism and Performance

F16DNF is not threaded in any implementation.

9 Further Comments

None.

10 Example

This example computes the largest component and index of that component for the vector

$$x = (1, 10, 11, -2, 9)^{\mathrm{T}}.$$

10.1 Program Text

```
    Program f16dnfe

!     F16DNF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: f16dnf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, incx, j, jx, k, n
!     .. Local Arrays ..
      Integer, Allocatable             :: x(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: abs
!     .. Executable Statements ..
      Write (nout,*) 'F16DNF Example Program Results'

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

      Read (nin,*) n, incx
      Allocate (x(1+(n-1)*abs(incx)))

!     Read the vector x and store forwards or backwards
!     as determined by incx.
      If (incx>0) Then
        jx = 1
      Else
        jx = 1 - (n-1)*incx
      End If

      Do j = 1, n
        Read (nin,*) x(jx)
        jx = jx + incx
      End Do
```

```
!     Find k = argmax(x) and i = max(x).

      Call f16dnf(n,x,incx,k,i)

      Write (nout,*)
      Write (nout,99999) k
      Write (nout,99998) i

99999 Format (1X,'Index of largest component of x is',I3)
99998 Format (1X,'Largest component of x is',I12)
    End Program f16dnfe
```

10.2 Program Data

```
F16DNF Example Program Data
  5   1                                                 : n and incx
  1
  10
  11
  -2
  9                                                     : Vector x
```

10.3 Program Results

```
 F16DNF Example Program Results

 Index of largest component of x is  3
 Largest component of x is          11
```