

# NAG Library Function Document

## nag\_imin\_val (f16dpc)

### 1 Purpose

nag\_imin\_val (f16dpc) computes the smallest component of an integer vector, along with the index of that component.

### 2 Specification

```
#include <nag.h>
#include <nagf16.h>

void nag_imin_val (Integer n, const Integer x[], Integer incx, Integer *k,
                  Integer *i, NagError *fail)
```

### 3 Description

nag\_imin\_val (f16dpc) computes the smallest component,  $i$ , of an  $n$ -element integer vector  $x$ , and determines the smallest index,  $k$ , such that

$$i = x_k = \min_j x_j.$$

### 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: | <b>n</b> – Integer   | <i>Input</i>  |
|    | <i>On entry:</i> $n$ , the number of elements in $x$ .   |               |
|    | <i>Constraint:</i> $n \geq 0$ .  |               |
| 2: | <b>x</b> [ <i>dim</i> ] – const Integer  | <i>Input</i>  |
|    | <b>Note:</b> the dimension, $dim$ , of the array <b>x</b> must be at least $\max(1, 1 + (n - 1) \times  \mathbf{incx} )$ .   |               |
|    | <i>On entry:</i> the vector $x$ . Element $x_i$ is stored in <b>x</b> [( $i - 1$ ) $\times$   <b>incx</b>  ], for $i = 1, 2, \dots, n$ .   |               |
| 3: | <b>incx</b> – Integer  | <i>Input</i>  |
|    | <i>On entry:</i> the increment in the subscripts of <b>x</b> between successive elements of $x$ .  |               |
|    | <i>Constraint:</i> <b>incx</b> $\neq$ 0.   |               |
| 4: | <b>k</b> – Integer *   | <i>Output</i> |
|    | <i>On exit:</i> $k$ , the index, from the set $\{0,  \mathbf{incx} , \dots, (n - 1) \times  \mathbf{incx} \}$ , of the smallest component of $x$ . If $n = 0$ on input then <b>k</b> is returned as $-1$ . |               |
| 5: | <b>i</b> – Integer *   | <i>Output</i> |
|    | <i>On exit:</i> $i$ , the smallest component of $x$ . If $n = 0$ on input then <b>i</b> is returned as 0.  |               |

6: **fail** – NagError \*

*Input/Output*

The NAG error argument (see Section 3.6 in the Essential Introduction).

## 6 Error Indicators and Warnings

### NE\_BAD\_PARAM

On entry, argument  $\langle value \rangle$  had an illegal value.

### NE\_INT

On entry, **incx** =  $\langle value \rangle$ .

Constraint: **incx**  $\neq 0$ .

On entry, **n** =  $\langle value \rangle$ .

Constraint: **n**  $\geq 0$ .

## 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

Not applicable.

## 9 Further Comments

None.

## 10 Example

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

$$x = (1, 10, 11, -2, 9)^T.$$

### 10.1 Program Text

```

/* nag_imin_val (f16dpc) Example Program.
 *
 * Copyright 2005 Numerical Algorithms Group.
 *
 * Mark 9, 2009.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagf16.h>

int main(void)
{
    /* Scalars */
    Integer exit_status, i, incx, j, k, n, xlen;
    /* Arrays */
    Integer *x = 0;
    /* Nag Types */
    NagError fail;

    exit_status = 0;
    INIT_FAIL(fail);

```

```

printf("nag_imin_val (f16dpc) Example Program Results\n\n");

/* Skip heading in data file */
scanf("%*[\n] ");
/* Read the number of elements and the increment */
scanf("%ld%ld%*[\n] ", &n, &incx);

xlen = MAX(1, 1 + (n - 1)*ABS(incx));

if (n > 0)
{
    /* Allocate memory */
    if (!(x = NAG_ALLOC(xlen, Integer)))
    {
        printf("Allocation failure\n");
        exit_status = -1;
        goto END;
    }
}
else
{
    printf("Invalid n\n");
    exit_status = 1;
    goto END;
}

/* Input vector x */
for (j = 0; j < xlen; j = j + incx)
    scanf("%ld", &x[j]);
scanf("%*[\n] ");

/* nag_imin_val (f16dpc).
 * Get minimum value (i) and location of that value (k)
 * of Integer vector */
nag_imin_val(n, x, incx, &k, &i, &fail);

if (fail.code != NE_NOERROR)
{
    printf("Error from nag_imin_val (f16dpc).\n%s\n", fail.message);
    exit_status = 1;
    goto END;
}

/* Print the minimum value */
printf("Minimum element of x is %12ld\n", i);
/* Print its location */
printf("Index of minimum element of x is %3ld\n", k);

END:
    NAG_FREE(x);

    return exit_status;
}

```

## 10.2 Program Data

```

nag_imin_val (f16dpc) Example Program Data
5      1                                : n and incx
1     10     11     -2     9           : Array x

```

## 10.3 Program Results

```

nag_imin_val (f16dpc) Example Program Results

Minimum element of x is          -2
Index of minimum element of x is    3

```

---