NAG Library Function Document

nag_iload (f16dbc)

1 Purpose
nag_iload (f16dbc) broadcasts a scalar into an integer vector.

2 Specification
#include <nag.h>
#include <nagf16.h>
void nag_iload (Integer n, Integer alpha, Integer x[], Integer incx, NagError *fail)

3 Description
nag_iload (f16dbc) performs the operation
\[ x \leftarrow (\alpha, \alpha, \ldots, \alpha)^T, \]
where \( x \) is an \( n \)-element integer vector and \( \alpha \) is an integer scalar.

4 References

5 Arguments
1: \( n \) – Integer
   \text{Input}
   \text{On entry:} n, the number of elements in \( x \).
   \text{Constraint:} n \geq 0.

2: \( \alpha \) – Integer
   \text{Input}
   \text{On entry:} the scalar \( \alpha \).

3: \( x[\text{dim}] \) – Integer
   \text{Output}
   \text{Note:} the dimension, \( \text{dim} \), of the array \( x \) must be at least \( \max(1, 1 + (n - 1)|\text{incx}|) \).
   \text{On exit:} the scalar \( \alpha \) is scattered with a stride of \( \text{incx} \) in \( x \). Intermediate elements of \( x \) are unchanged.

4: \( \text{incx} \) – Integer
   \text{Input}
   \text{On entry:} the increment in the subscripts of \( x \) between successive elements of \( x \).
   \text{Constraint:} \text{incx} \neq 0.

5: \( \text{fail} \) – NagError *
   \text{Input/Output}
   The NAG error argument (see Section 3.6 in the Essential Introduction).
6 Error Indicators and Warnings

**NE_ALLOC_FAIL**
Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction for further information.

**NE_BAD_PARAM**
On entry, argument \(\text{value} \) had an illegal value.

**NE_INT**
On entry, \(\text{incx} = \langle\text{value}\rangle\).
Constraint: \(\text{incx} \neq 0\).
On entry, \(\text{n} = \langle\text{value}\rangle\).
Constraint: \(\text{n} \geq 0\).

**NE_INTERNAL_ERROR**
An unexpected error has been triggered by this function. Please contact NAG.
See Section 3.6.6 in the Essential Introduction for further information.

**NE_NO_LICENCE**
Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in the Essential Introduction for further information.

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
See Section 10 in nag_dgeqpf (f08bec) and nag_zgeqpf (f08bsc).