

NAG C Library Function Document

nag_rngs_permute (g05nac)

1 Purpose

nag_rngs_permute (g05nac) performs a pseudo-random permutation of a vector of integers.

2 Specification

```
void nag_rngs_permute (Integer index[], Integer n, Integer igen, Integer iseed[],
                      NagError *fail)
```

3 Description

nag_rngs_permute (g05nac) permutes the elements of an integer array without inspecting their values. Each of the $n!$ possible permutations of the n values may be regarded as being equally probable.

Even for modest values of n (greater than 25 say), it is theoretically impossible that all $n!$ permutations may occur, as $n!$ exceeds the cycle length of nag_rngs_basic (g05kac) for any valid value of **igen**. For practical purposes this is irrelevant, as the time necessary to generate all possible permutations is many millenia.

One of the initialisation functions nag_rngs_init_repeatable (g05kbc) (for a repeatable sequence if computed sequentially) or nag_rngs_init_nonrepeatable (g05kcc) (for a non-repeatable sequence) must be called prior to the first call to nag_rngs_permute (g05nac).

4 References

Knuth D E (1981) *The Art of Computer Programming (Volume 2)* (2nd Edition) Addison–Wesley

Kendall M G and Stuart A (1969) *The Advanced Theory of Statistics (Volume 1)* (3rd Edition) Griffin

5 Parameters

- 1: **index**[**n**] – Integer *Input/Output*
On entry: the n integer values to be permuted.
On exit: the n permuted integer values.
- 2: **n** – Integer *Input*
On entry: the number of values to be permuted.
Constraint: $n \geq 1$.
- 3: **igen** – Integer *Input*
On entry: must contain the identification number for the generator to be used to return a pseudo-random number and should remain unchanged following initialisation by a prior call to one of the functions nag_rngs_init_repeatable (g05kbc) or nag_rngs_init_nonrepeatable (g05kcc).
- 4: **iseed**[4] – Integer *Input/Output*
On entry: contains values which define the current state of the selected generator.
On exit: contains updated values defining the new state of the selected generator.
- 5: **fail** – NagError * *Input/Output*
The NAG error parameter (see the Essential Introduction).

6 Error Indicators and Warnings

NE_INT

On entry, $n = \langle value \rangle$.
Constraint: $n \geq 1$.

NE_BAD_PARAM

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

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please consult NAG for assistance.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

In the example program a vector containing the first eight positive integers in ascending order is permuted by a call to `nag_rngs_permute (g05nac)` and the permutation is printed. This is repeated a total of ten times, after initialisation by `nag_rngs_init_repeatable (g05kbc)`.

9.1 Program Text

```

/* nag_rngs_permute(g05nac) Example Program.
 *
 * Copyright 2001 Numerical Algorithms Group.
 *
 * Mark 7, 2001.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg05.h>

int main(void)
{
    /* Scalars */
    Integer i, igen, j, k, m, n;
    Integer exit_status=0;
    NagError fail;

    /* Arrays */
    Integer *index=0;
    Integer iseed[4];

    INIT_FAIL(fail);
    Vprintf("g05nac Example Program Results\n\n");

    n = 8;
    m = 10;
    /* Allocate memory */
    if ( !(index = NAG_ALLOC(n, Integer)) )
    {
        Vprintf("Allocation failure\n");
        exit_status = -1;
    }
}

```

```

    goto END;
}

/* Initialise the seed to a repeatable sequence */
iseed[0] = 1762543;
iseed[1] = 9324783;
iseed[2] = 42344;
iseed[3] = 742355;
/* igen identifies the stream. */
igen = 1;
g05kbc(&igen, iseed);
Vprintf("%2ld Permutations of first %lld integers\n", m, n);
Vprintf("\n");

for (j = 0; j < m; ++j)
{
    for (i = 0; i < n; ++i)
        index[i] = i + 1;
    g05nac(index, n, igen, iseed, &fail);
    if (fail.code != NE_NOERROR)
    {
        Vprintf("Error from g05nac.\n%s\n", fail.message);
        exit_status = 1;
        goto END;
    }
    for (k = 0; k < n; ++k)
    {
        Vprintf("%3ld%s",
                index[k], (k+1)%8 == 0 || k == n-1 ? "\n": " ");
    }
}
END:
if (index) NAG_FREE(index);
return exit_status;
}

```

9.2 Program Data

None.

9.3 Program Results

g05nac Example Program Results

10 Permutations of first 8 integers

2	4	3	7	8	6	5	1
2	3	6	4	7	5	1	8
6	5	1	4	7	8	3	2
7	3	2	5	1	4	8	6
7	4	8	5	6	2	3	1
2	1	5	8	4	3	7	6
7	3	4	6	2	5	1	8
4	1	7	5	6	8	3	2
3	1	5	2	8	7	4	6
8	7	6	1	3	2	5	4
