nag_save_random_state (g05cfc)

1. Purpose

nag_save_random_state (g05cfc) saves the value of the seed used by the basic generator in the g05 Chapter.

2. Specification

```
#include <nag.h>
#include <nagg05.h>

void nag_save_random_state(Integer istate[], double xstate[])
```

3. Description

This function saves information about the basic generator to enable nag_restore_random_state (g05cgc) subsequently to restore the basic generator to its current state. The values of **istate** and **xstate** must not be altered between a call of nag_save_random_state and a call of nag_restore_random_state (g05cgc).

4. Parameters

istate[**9**]
xstate[**4**]
Output: information about the generator.

5. Error Indications and Warnings

None.

6. Further Comments

None.

7. See Also

nag_random_continuous_uniform (g05cac)
nag_restore_random_state (g05cgc)

8. Example

The program prints 10 pseudo-random numbers generated by nag_random_continuous_uniform (g05cac); it saves the generator state after the 2nd, and restores it after the 7th so that the 8th, 9th and 10th numbers are the same as the 3rd, 4th and 5th.

8.1. Program Text

```
/* nag_save_random_state(g05cfc) Example Program
 *
 * Copyright 1990 Numerical Algorithms Group.
 *
 * Mark 1, 1990.
 */

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

main()
{
  Integer seed = 0;
```

```
  Integer i, istate[9];
  double x[5], xstate[4];

  Vprintf("g05cfc Example Program Results\n");
  g05cbc(seed);
  for (i= 0; i<5; ++i)
    {
      x[i] = g05cac();
      if (i == 1)
        g05cfc(istate, xstate);
    }
  for (i=0; i<5; ++i)
    Vprintf("%9.4f%s", x[i], (i%5==4 || i==4) ? "\n": " ");
  for (i=0; i<5; ++i)
    {
      x[i] = g05cac();
      if (i == 1)
        g05cgc(istate, xstate, NAGERR_DEFAULT);
    }
  for (i=0; i<5; ++i)
    Vprintf("%9.4f%s", x[i], (i%5==4 || i==4) ? "\n": " ");
  exit(EXIT_SUCCESS);
}
```

8.2. Program Data

None.

8.3. Program Results

```
g05cfc Example Program Results
   0.7951    0.2257    0.3713    0.2250    0.8787
   0.0475    0.1806    0.3713    0.2250    0.8787
```