NAME

DR250 – Generate a vector of double-precision pseudorandom numbers.

SYNOPSIS

CALL DR250(N,X)

N	is the INTEGER*4 number of values to be generated, or a seed
X(N)	is the REAL*8 vector of random numbers returned

DESCRIPTION

The generator is initialized at load time with values that are suitable for many applications. If $N \le 0$, the generator is dynamically reinitialized with values obtained using the linear congruential generator described in [1]. If $N < 0$ on input, the seed used for the reinitialization process is $|N|$ and on return N contains the negative of the final new seed resulting from the initialization process. If $N = 0$ on input, the seed is obtained from the time-of-day clock and on return N contains the negative of that seed.

If $N > 0$, the generalized shift-register algorithm described in [2] is used to generate N double-precision values in X, and N is not changed. When DR250 is called repeatedly in the same program with $N > 0$, it generates different vectors; the sequence of values that it returns does not start over from the beginning unless it is reinitialized by being called with $N < 0$ as described above.

The shift register described in [2] is internal to DR250 rather than being part of X, and is inaccessible to the calling program. The values returned in X are normalized IEEE double-precision floating-point numbers uniformly distributed on the interval [0,1]. For most starting seeds, the period of repetition of the generator is $2^{250} - 1$, which is approximately $1.8 \times 10^{75.}$

SEE ALSO

MCRNG, which generates a sequence of INTEGER*4 pseudorandom numbers
RNDBYT, which generates a sequence of pseudorandom bytes
Linux routines rand(1), random(4), urandom(4)

BUGS

Precautions are necessary to avoid small systematic errors when shift-register generators such as DR250 are used with the Wolff algorithm for solving Ising spin models [3].

LINKAGE

```
gfortran   source.f   –L${HOME}/lib   –lmisc
```

AUTHOR

Michael Kupferschmid

REFERENCES

[1] Forsythe, G., Malcolm, M., Moler, C., Computer Methods for Mathematical Computations, Prentice-Hall (1977) 246.
[2] Kirkpatrick, Scott and Stoll, Erich P., "A Very Fast Shift-Register Sequence Random Number Generator," Journal of Computational Physics 40 (1981) 517-526.
[3] Ferrenberg, Alan M., Landau, D. P., Wong, Y. Joanna, "Monte Carlo Simulations: Hidden Errors from 'Good' Random Number Generators" Physical Review Letters 69 (1992) 3382-3384.

EXAMPLE

```
      REAL*8 X(10)
      INTEGER*4 SEED/-1982637093/
      CALL DR250(SEED,X)
      WRITE(6,901) SEED
  901 FORMAT('Next SEED = ',I10)
      CALL DR250(10,X)
      WRITE(6,902) X
  902 FORMAT(F18.16)
      STOP
      END
```

This example produced the following output:

```
unix[1]
Next SEED = -951968321
0.4879140041744168
0.1355766636529716
0.8869387106275517
0.6143979461413793
0.5597671176048773
0.5186943224153766
0.3863282290103363
0.3538381107073243
0.3968800583858356
0.8663204197641670
unix[2]
```