

Exercise 7 FYSY160 C++ numeerinen ohjelmointi Autumn 2011

1. Write your own C++ program, that produces normal (gaussian) distributed random numbers with variance 1. Numbers are produced in the function `GaussRand()`.

Usage:

```
double GaussRand(void);
```

returns a normal distributed random number of type `double` .

Here is the pseudocode for `GaussRand`

(algorithm: J. F. Fernández and J. Rivero, *Comput. Phys.* 10, 83 (1996))

```
// constants
// =====
nk = 16
nregs = 1024
ntherm = 10*nregs
nstore = 100
// initialization
// =====
if(called for the first time)
  // precompute sin and cos
  do k = 1, nk
    angle = random angle between 0...2*pi
    ck(k) = cos(angle)
    sk(k) = sin(angle)
  end do
  // thermalize contents of registers r()
  r(1:nregs)=1 // init all registers to 1
  m = 0
  do i = 1, ntherm
    m=(m+1)%nregs
    pick random register n (/= m)
    update_registers n and m
  end do
  nused = nstore // mark all used
end if
// production loop
// =====
if(nused = nstore) // store empty
  // fill store with nstore gaussian random numbers
```

```

m = 0
do i = 1, nstore, step 2
  m=(m+1)%nregs
  pick random register n (/= m)
  update_registers n,m
  store(i) = r(n) // first gaussian random number
  store(i+1) = r(m) // second gaussian random number
end do
nused = 0 // nothing used yet
else
  // use a number in store
  nused = nused + 1
  return store(nused) as result
end if

```

Registers are updated like this:

```

pick random k between 0...nk-1
x = r(m)*ck(k)+r(n)*sk(k)
r(n) = -r(m)*sk(k)+r(n)*ck(k)
r(m) = x

```

Hints:

- You can use less random `rand()` to get random numbers; (in real life you should use either a generator function given in Boost or in GSL, not required here).
- One way to get a crude random integer `k` between `0... kmax` is `k=rand()%(kmax+1)` (works if `kmax < RAND_MAX`)
- I recommend you use `vector<double>` containers, not dynamical tables.
- Many variables need to be kept `static` (stored in memory permanently between function calls), such as `store` and `nused` (store usage) *etc.* Initialization part is done only when function is called for the first time. You can set a variable `static first` to achieve this.
- Reorganize the code as much as you wish.