An exercise for Dr.R.M.Burstall.

Dear Rod,

because --as you know-- we Dutch are a God-fearing nation, Ascension-day is here an official Holiday, and on official Holidays I don't work. Today I just fooled with figures.

In doing so I discovered a function of the natural numbers wich has a nice recursive definition, viz.

fusc(1) = 1
fusc(2n) = fusc(n)
fusc(2n+1) = fusc(n) + fusc(n+1)

a definition which, as far as complexity is concerned, seems to lie between the Fibonacci series and the Pascal triangle.

(The function fusc is of a mild interest on account of the following property: with f1 = fusc(n1) and f2 = fusc(n2) the following two statements ~~are~~ hold: ~~equivalent:~~
"if there exists an N such that $n1 + n2 = 2^N$, then f1 and f2 are relatively prime" and "if f1 and f2 are relatively prime, then there exist an n1, an n2, and an N, such that $n1 + n2 = 2^N$. In the above recursive definition, this is no longer obvious, at least not to me; hence its name.)

Having seen your exercises concerning the derivation of an iterative program, starting with the recursive definition for the n-th number of the Fibonacci series, I was suddenly reminded of that exercise when I was considering an iterative program for the computation of fusc. It should be a rewarding exercise, as there exists a very nice iterative program:

```
n, a, b := N, 1, 0;
do n ≠ 0 and even(n) → a, n:= a + b, n/2
 [] odd(n) → b, n:= b + a, (n-1)/2
od {b = fusc(N)}
```

I wish you luck and enjoyment! Yours ever, Edsger

Plataanstraat 5
NL-4565 NUENEN
The Netherlands

prof.dr.Edsger W.Dijkstra
Burroughs Research Fellow

Elemente der Mathematik V.2 p.95 (1947) de Rahm