Exercise 1.43.

If f is a numerical function and n is a positive integer, then we can form the nth repeated application of f, which is defined to be the function whose value at x is $f\,(f\,(\ldots(f\,(x))\ldots))$. For example, if f is the function $x \mapsto x+1$, then the nth repeated application of f is the function $x \mapsto x+n$. If f is the operation of squaring a number, then the nth repeated application of f is the function that raises its argument to the 2^nth power. Write a procedure that takes as inputs a procedure that computes f and a positive integer n and returns the procedure that computes the nth repeated application of f. Your procedure should be able to be used as follows:

```
((repeated square 2) 5)
625
```

Hint: You may find it convenient to use `compose` from exercise 1.42.

Answer.

Before setting out to design the procedure, let's first analyze the structure of the expression $f\,(f\,(\ldots(f\,(x))\ldots))$. Being defined recursively, the function $x \mapsto f\,(f\,(\ldots(f\,(x))\ldots))$ can be evaluated in a way that

i. If n equals 1, then just return f;

ii. otherwise, compose the function f with the inner function which bears an identical form.

Now we can write the `repeated` procedure using the strategy described above:

```
(define repeated
  (lambda (f n)
    (if (= n 1)
        f
        (compose f
                 (repeated f (- n 1))))))
```