I plugged this function into the scam interpreter and ran it with input of 2 which resulted in a segmentation fault. What went wrong here? As for the answer to the question, is this not an infinite loop? I believe the function is an iterative function - is that correct?

```
scam> (define (f n)
    more> (f (+ n 1))
    more> 0
    more> )
<function f(n)>
scam> (f 2)
Segmentation fault: 11
```

What SHOULD happen is you hit the stack limit and get an "infinite recursion?" message, since the recursive call is not tail recursive. Another student reported a similar bug on another thread. I'm looking into it...

ok thank you! I must have missed that other thread - my mistake.

Kudos to J. Robinson for figuring out what was going wrong. It turns out that the Scam stack was set larger than the underlying C stack. So when the C stack ran out of space, you got a segmentation fault. The temporary fix is to set your C stack larger. On Linux, something like:

```
ulimit -s 16000
```

will do. I don't know what to do on a Mac.