MapReduce
Talked about MapReduce for a little bit

Lazy Programmers are the Best Programmers
What's printed at each line?

1.
> (define x (+ 2 3))
> x ==> 5
> (define y ((lambda(a) a) (* 3 4)))
> y ==> 12
> (define z ((lambda(b) (+ b 10)) y))
> z ==> 22
Note that nothing different happens in the lazy evaluator if we don’t use set!

2.
> (define count 0)
> (define (foo x y) (x y))
> (define z (foo (lambda(a) (set! count a) (* a a))
>                  (begin (set! count (+ 1 count)) count)))
The last define will cause an infinite loop.

3.
> (define count 0)
> (define (incr!) (set! count (+ count 1)))
> (define (foo x)
>     (let ((y (begin (incr!) count)))
>       (if (<= count 1)
>           (foo y)
>           x))
> (foo 10) ==> infinite loop

4. Suppose that the following expressions are entered in the lazy evaluator:

> (define (truth x y) (display (+ x 1)) y)
> (define beauty (truth (* 6 7) (- 5 2)))
> beauty

Which line will cause + to be executed? *? - ? B, B, C