**On the board, like the previous one:**

(define square-sum

(lambda (n)

(if (zero? n)

0

(+ (\* n n) (square-sum (- n 1))))))

**This one is on the quiz. Let them have about 3 minutes to work on it.   
Ask if everyone has it. If so, then move on, and don't go over it.**

(define square-all

(lambda (ls)

(if (null? ls)

'()

(cons (\* (car ls) (car ls))

(square-all (cdr ls))))))

**On the board, like the previous one:**

(define make-list

(lambda (n obj)

(if (zero? n)

'()

(cons obj (make-list (- n 1))))))

(define firsts

(lambda (ls)

(if (null? ls)

'()

(cons (car (car ls))

(firsts (cdr ls))))))

(define firsts

(lambda (ls)

(map car ls)))

(define square-all

(lambda (ls)

(map (lambda (n) (\* n n))

ls))))