Today

- Copies (we need volunteers)
- Count-evens
- Find-evens
- Recursion In Two Directions

Copies

(define (copies n wd)
  (if (< n 1)
      '()
      (sentence wd (copies (- n 1) wd)))))

Count the number of even-numbers

(define (count-evens sent)
  (cond ((empty? sent) 0) ; empty?
        ((even? (first sent)) ; base case: return 0
         (+ 1 (count-evens (bf sent)))) ; recurse on the rest of sent
        ((odd? (first sent))
         (+ 0 (count-evens (bf sent))) ; recurse on the rest of sent
         )))

The Leap of Faith...

Dude this seems like a hard problem! I'll do this small piece and hope that someone can do the rest.

> (find-evens '(2 3 4 5 6))

(2 4 6)
Problem: find all the even numbers in a sentence of numbers

(define (find-evens sent)
  (cond [(empty? sent)  ; base case
         ()
        ; rec case 1
        [(odd? (first sent)) (find-evens (bf sent))]
        [else  ; rec case 2: even
         (se (first sent) (find-evens (bf sent)))]
    ))

The Leap of Faith...

Dude this seems like a hard problem!
I'll do this small piece and hope that someone can do the rest.

- Sum-in-interval
  - What was the “small piece”?
  - What was the “rest”?