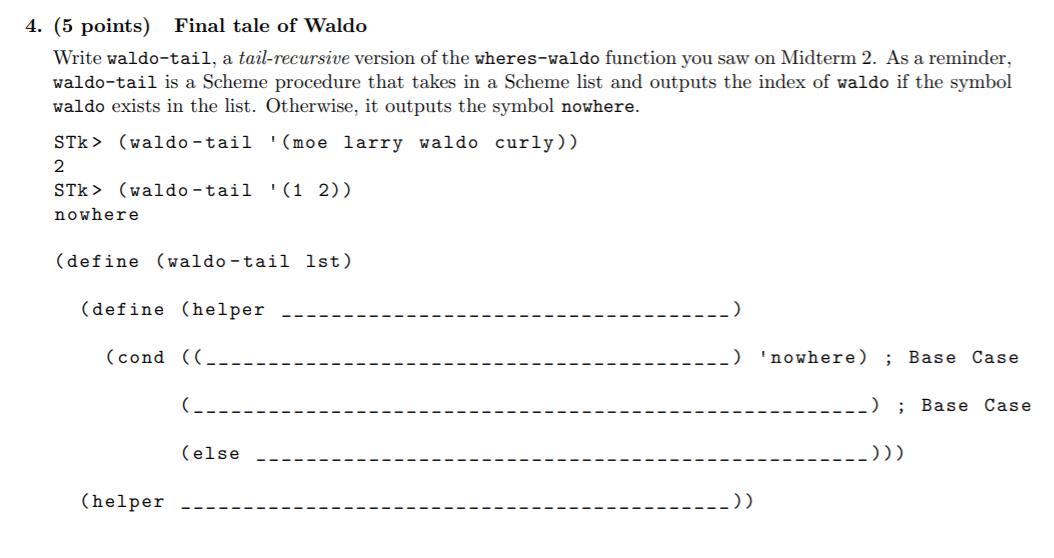
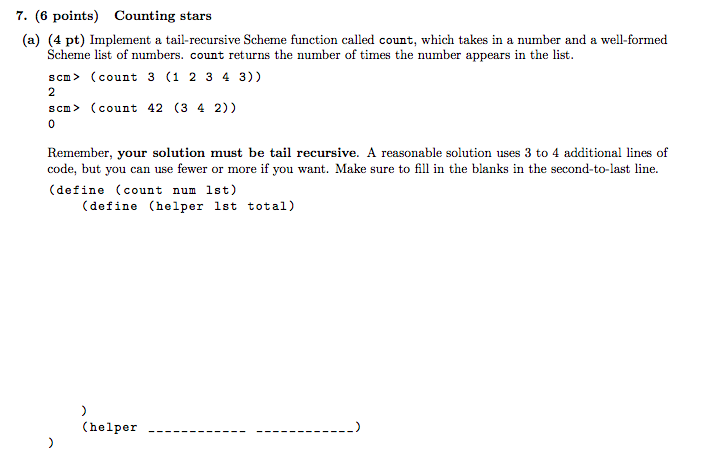
**Spring 2019 Tail Recursion Topical Review**

Mary & Chae

**Q1**:



**Q2**:



**Q3:**

Implement **sum-satisfied-k** which, given an input list **lst**, a predicate procedure **f** which takes in one argument, and an integer **k**, will return the sum of the first **k** elements that satisfy **f**. If there are not k such elements, return 0. Make sure your implementation is tail recursive!

(**define** (sum-satisfied-k lst f k)

(**define** (sum-helper \_\_\_\_\_\_\_\_\_\_\_\_\_\_ )

(**cond** ((= k 0) \_\_\_\_\_\_\_\_\_ )

((null? lst) \_\_\_\_\_\_\_\_\_\_\_)

((\_\_\_\_\_\_\_\_\_\_\_\_)(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_))

(else \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_))

(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

)

**Q4:**

Implement **remove-range** which, given one input list **lst**, and two integers **i** and **j**, returns a new list containing the elements of lst in order, minus the elements from i to j inclusive. For example, given the list (0 1 2 3 4), with i = 1 and j = 3, we would return the list (0, 4). Your implementation must be tail recursive. You may assume j >= i, i is non-negative, and j is less than the length of the list. (Hint: You may want to use the built-in append function)

(**define** (remove-range lst i j)

(**define** (remove-tail \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(**cond** ((\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_))

((\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ))

(**else** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_))

( \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

)

