Implement the following Racket functions:

1. A function called **C** that implements the function of type $\mathbb{N} \to \mathbb{N}$ defined by the equation

$$C(1) = C(2) = 1$$

$$C(n) = C\left(C(n-1) + C\left(n - C(n-1)\right)\right)$$

2. A function called **A** that implements the function of

$$A(1) = A(2) = 1$$

$$A(n) = A(A(n-1)) + A(n - A(n-1))$$

3. Implement a scheme predicate called **set-equal?** that takes as input two lists representing sets and returns true if the lists represent the same set and false otherwise. The difference between this predicate and the one discussed in detail in class is that this predicate will support sets of sets to any degree of nesting, e.g., sets of sets of sets. Thus, the call

   (set-equal? '(a (b (c d))) '(((d c) b) a))))

   should return **#t** and the call

   (set-equal? '(a (b (c d))) '((d c b) a))))

   should return **#f**.

Be sure to name the functions exactly as indicated as we will be testing it with a script.

This lab will be turned in by posting a file with the Racket definitions of all necessary functions to Canvas.