**Directions**: You can use the Scheme `atom?` procedure without writing it. Any other helper functions you need that aren’t a standard part of Scheme you should write yourself. Helper functions can be external; they don’t need to be defined with `letrecs`. There are 5 programming questions here and 2 “explain in one or two sentences” questions. I put the latter at the end because they are different; don’t run out of time and not get to them. All 7 questions are equally weighted. Remember to write the Honor Pledge on the last page when you have finished the exam.

1. Write procedure `(remberSecond a lat)` which removes the second occurrence of atom `a` in `lat`.  
   So `(remberSecond ‘a ‘(a b c a b c a))` returns `(a b c b c a)` and 
   `(remberSecond ‘a ‘(a b o b))` returns `(a b o b)`
2. Here is the fold function:
(define fold
  (lambda (recur base lat)
    (letrec ([h (lambda (s)
                  (cond
                    [(null? s) base]
                    [else (recur (car s) (h (cdr s)))]))])
      (h lat))))

Use fold to write (count a lat) which returns the number of times atom a occurs in lat.
(count ‘a ‘(a b r a c a d a b r a ) ) returns 5
3. Now use map and apply to write (count* a L) which returns the number of times atom a occurs in general list L. (count* ‘a ‘(a b (r (a c) a (d (a b (r a))))) returns 5.
4. Suppose we have a list of pairs of atoms and numbers representing people and their ages, such as ‘((mary 18) (tom 16) (bob 62) (sylvia 23))’ Give a Scheme function (oldest age-list) that returns the atom representing the oldest person in the list. (oldest ‘((mary 18) (tom 16) (bob 62) (sylvia 23))’) should return ‘bob.'
5. An environment associates values with symbols; give it a symbol and it returns the value bound to that symbol. I want to represent environments with procedures, so if I have symbol x bound to 3 in env, then (env ‘x) returns 3. If y is not bound to anything in env then (env ‘y) returns ‘error. Write procedure (extend-env syms vals old-env) that extends old-env with bindings of syms to vals and returns the new environment. For example if we create new-env as (define new-env (extend-env ‘(x y z) ‘(1 2 3) old-env)) then (new-env ‘x) returns 1, (new-env ‘y) returns 2 and (new-env ‘z’) returns 3.
6. Explain in one or two sentences what it means for a procedure to have state. Give an example of a procedure with state and an example of one without state.
7. We said the Y-combinator is

(define YC
  (lambda (a)
    (lambda (F) (F F))
    (lambda (f)
      (a (lambda (x) ((f f) x))))))

This may be the wackiest definition you have ever seen. Explain in one or two sentences what the significance of the Y-combinator is.