

CS 275
Exam I
October 10, 2019

Directions: There are 7 numbered problems worth 14 points each; you get 2 points for free. You don't need to write helper functions via `letrecs`; you can define them at the top level (and that might make your code easier to read). You can assume there is a predicate `(atom? x)` and another one `(number? x)`; you don't need to write those. **Any other helper functions you should write yourself.** In the following questions argument *lat* is always a flat list, such as `'(a b c)` and argument *L* is a general list, such as `'(a (b c (d e)) f)`. There is a place on the last page for you to sign the Honor Pledge.

1. Write procedure **(duplicate a lat)** that adds a second instance of atom *a* every time *a* is found in the flat list *lat*. For example, `(duplicate 'b '(a b a c a b a))` returns `(a b b a c a b b a)`

2. Write procedure **(removeDuplicates lat)**. As usual lat is a flat list of atoms. For each run of identical entries in lat, such as the 3s in (1 2 3 3 3 2 1), this procedure will remove all but one of those entries. So (removeDuplicates '(1 3 3 3 3 4 2 2 1)) returns (1 3 4 2 1), and (removeDuplicates '(1 2 1 2 3)) returns (1 2 1 2 3)

3. Use `foldr` or `foldl` to write **(count a lat)** which returns the number of instances of atom `a` in `lat`, a flat list of atoms. For example, `(count 3 '(1 2 3 2 3 2 3 4 3 3))` returns 5

4. Consider the following function:

```
(define B
  (lambda (L)
    (cond
      [(null? L) null]
      [(atom? L) (if (eq? L 'bob) (list L) null)]
      [else (apply append (map B L))])))
```

a) What is (B '(1 2 3 bob))?

b) What is (B '((1 3 bob (4)) (5 ((6))) (7 (8 bob) 9))) ??

5. What does the following expression evaluate to in the top-level environment? Be very explicit:

```
(let ([a 5] [b 3])  
      (lambda (x y) (* a (+ b (* x y)))))
```

6.

a. Write **(last lat)** which returns the last atom in the flat list lat. For example, (last '(a b c d)) returns d. None of the entries of lat will be null.

b. Write **(last* L)** which returns the last non-null atom in the general list L. For example, (last* '(a (b (c)) (d (e f)) (())) should return f.

7. Write function (**separateNums L**) that returns a list of two flat lists: one containing the numbers of L, the other containing any other atoms of L. Both lists should have their atoms in the same order as L. For example,
(separateNums '(a b 3 c 4 2 d 5)) returns ((3 4 2 5) (a b c d)) while
(separateNums '((a b (c (d 1 2) 3)) ((e 4 5 (f))))) returns ((1 2 3 4 5) (a b c d e f))

You can use this page as extra room for any problem. If you want me to grade something here be clear about which problem it refers to.

Please write and sign the honor pledge when you are finished with the exam.