CSE3401 Summer 2012 Assignment #1, Due by May 27th 2012 11:59PM

Exercise 1 (5 points)

Define the following function in your LISP interpreter session:

```
(defun make_atom_or_list (level)
        (if (or (zerop level) (and (< level 2) (zerop (random 2))))
                (random 10)
                (mapcar #'(lambda (x) (make_atom_or_list (1- level)))
                        (make-list (random 5))
                )
        )
)
```

Then evaluate the following form a few times:

```
(make_atom_or_list 4)
```

This creates a random list of lists. Pick one non-trivial and draw the representation.

Exercise 2 (10 points)

Write a recursive LISP function to test if a given s-expression is a proper LISP list and not just a dotted-pair expression that has no list representation. For example:

(listpp '(a b c))
➔ (a b c)

(listpp '(a b . c))
➔ nil

(listpp '(a (b . c) d))
➔ nil

Note: Use comments to make your code clear to understand. Perform thorough testing of your functions and include in your answer. You can use helper functions. Be careful how you handle NILs since they are lists but not conses.

Exercise 3 (10 points)

Write a function that can compare 2 s-expressions and output T if their structures are equivalent (similar to the EQUAL predicate). You are not allowed to use the EQUAL predicate but you can freely use the EQL predicate.

NOTE: s-expressions cannot produce circular structures so do not worry about structures

Exercise 4 (10 points)

Write a function CIRCLIST that given a list will produce a circular queue by linking its tail back to the beginning of the given list.

Write an s-expression that uses CIRCLIST and converts the following nested list (a b (e f g h) c d) into 2 linked circular queues

Exercise 5 (15 points)

Write a function CIRCEQUAL that can compare 2 Lisp structures (including those that have cycles) and output T if their structures are equivalent.

Exercise 6 (10 points)

In the MAKE_ATOM_OR_LIST function defined in exercise 1, explain clearly and in detail:

- What does the IF conditional do for use here
- What does the MAPCAR invocation do for us here
- Where are the 3 elements of successful recursive function definitions