CS 331 – Exam Review Problem(s) – February 5

Submit your solution using the D2L "blank quiz" for this set of review problems before noon on the day of the next class meeting

In class on Wednesday we developed definitions of three functions – isMember, removeFirst, and subst. The definitions we developed in class closely resembled what you see below.

```
1  var isMember = function (a, lnums) {
2      if (isNull(lnums)) {
3          return false;
4      } else {
5          return isEq(a, car(lnums)) || isMember(a, cdr(lnums));
6      }
7  }
8
9  var removeFirst = function (x, l) {
10     if (isNull(l)) {
11         return [];
12     } else if (isEq(x, car(l))) {
13         return cdr(l);
14     } else {
15         return cons(car(l), removeFirst(x, cdr(l)));
16     }
17 }
18
19 var subst = function (n, o, l) {
20     if (isNull(l)) {
21         return [];
22     } else if (isEq(o, car(l))) {
23         return cons(n, subst(n, o, cdr(l)));
24     } else {
25         return cons(car(l), subst(n, o, cdr(l)));
26     }
27 }
```

1. Suppose we define the isMember function as follows:

   ```
   var isMember = function (a, lnums) {
       if (isEq(a, car(lnums))) {
           return true;
       } else if (isMember(a, cdr(lnums))) {
           return true;
       } else {
           return false;
       }
   }
   ```

   Which of the following best characterizes this new version of the function?

   (a) Stylistically it is not as good, but it does work reliably
   (b) It could never return true
   (c) It could never return false

2. Suppose that we wanted removeFirst to remove *all* occurrences of x in l instead of just the first. This could be accomplished by making a small change in one line and leaving the rest of the function unchanged. Which line is it? (You will be asked to explain what the change is in class.)

   (a) 10
   (b) 11
   (c) 12
   (d) 13
   (e) 15

3. Suppose that we want to make the *subst* function above do "deep substitution" at all levels of a list. For example:

   ```
   > subst(2, 3,[3, 9, [4, 2, 3], 3, 5])
   [ 2, 9, [ 4, 2, 2 ], 2, 5 ]
   ```

   Which of the following best characterizes how this could be done? (You will be asked to justify your choice in class.)

   (a) Your would need to add one additional *else-if* with one recursive call to *subst*
   (b) Your would need to add one additional *else-if* with two recursive calls to *subst*
   (c) Your would need to add two additional *else-if*s, each with one recursive call to *subst*