AI3 -- Knowledge Representation and Inference I

Tutorial exercise for Friday 23rd October 1987

1. Find the sequences of cars and cdrs that return x when applied to the following lists:

   a) (a b x d)
   b) (a (b (x d)))
   c) (((a (b (x) d))))

2. Write a LISP function called switch that accepts a two-element list and returns a list with these elements in the opposite order. For example, (switch '(a b)) should return (b a).

3. Write a LISP function multiple-member that returns t if its first argument occurs at least twice in its second. Remember that (member 'b '(a b c d)) returns (b c d).

The LISP function member returns checks to see if its first argument occurs at the "top level" of its second. That is, (member 'a '(b (d a) c)) returns nil. Write a recursive function which does this.

5. Write a LISP function deep-member which checks to see if its first argument occurs in its second at ANY level. That is, (deeper-member 'a '(b (d a) c)) should return a non-nil value.

6. Write a LISP function which removes all occurrences of an element from a list.

7. Write a LISP function to compute the intersection of two lists. No element should appear repeatedly in the result even if it appears repeatedly in the arguments.