**Moti Begna**

**CSCI 5106**

**Homework 8**

**Problem 1**

Using only the *append* relation, formulate queries to determine the following:

a) The third element of a list

Solution: append([\_, \_, X|\_], [\_], L).

b) The last element of a list.

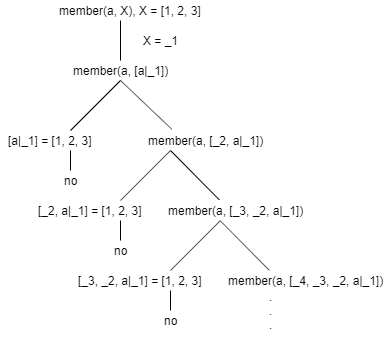
Solution - append([\_|\_], [X], List).

**Problem 2**

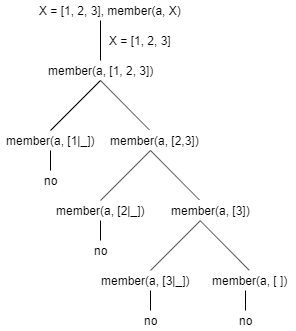
**Problem 3**

In part a, the prolog query first assigns the variable X to a specified list, and then runs a member relation on the value ‘a’ and X, while part b does this in the opposite direction. Based on the definitions of the member relations from the exercises, the member query in part a would match the value ‘a’ to the head of the list X, and then recurse over the tail of the list until matching against an empty list. In each instance, the match would fail since ‘a’ does not exist within the list X. In any case, the query as a whole would terminate. This greatly differs from part b, however, since the list X is constantly being reassigned. Here, a call to member on the value ‘a’ and X assigns the list X, firstly, as a list with ‘a’ as it’s head. Then, instead of the query X = [1, 2, 3] assigning the list to X, it would check to see if the list is *equal* to X. And because this would end up failing, a call to member would once again be called, only this time, ‘a’ becomes the second element of X. This procedure would run indefinitely, constantly assigning ‘a’ to the next index of the list X, and then checking its equality with the list [1, 2, 3]. This is due to the fact that X is not assigned when member(a, X) is called, thus X can be any list so long as a exists somewhere within it.

**Part a)**

****

**Part b)**

****