In this small problem set we were supposed to experiment with prolog and use list traversals to answer two problems. The first problem was to provide a unique list of leaves of a given binary search tree. The other problem was to report the Longest path from the root to a leaf of a Binary search tree, given these 3 rules.

1) Mydepth of nil is 0.
2) Mydepth of an atom is 0
3) Mydepth of a list is the max(depth(H), depth(T)) + 1

Problem 1

```
% Name: Marco Salazar
% Date: 11/30/2020
% Assignment: Practice with Prolog Programming
% Problem: When given a binary tree as a list, I need to output a unique list of the leaves of
that tree.
% eg. mytreeunique([a,[b,[a,[c,d]]]],X).
% Yields X=[a,b,c,d]

% flattening a list
flatten([], []).
flatten(X, [X]) :- atom(X),!.

flatten([H|T], Z) :- flatten(H, T1), flatten(T, T2), append(T1, T2, Z).

% we assume we always get two lists for append
append([],L,L).
append([H|T], L, [H | Z]) :- append(T,L,Z).

% make sure that a list has unique elements
myuniq([], []).
myuniq([H | T], L) :- member(H, T),!, myuniq(T,L).
myuniq([H | T], [H | L] ) :- myuniq(T, L).

% given a binary tree as a list, this provides the unique list of leaves of the tree
mytreeunique([],[]).
mytreeunique(X, Z) :- flatten(X, Y), myuniq(Y, Z).
```

```
?- ['prolog.mytreeunique'].
true.

?- mytreeunique([a,[b,[a,[c,d]]]],X).
X = [b, a, c, d] .

?- mytreeunique([],X).
X = [] .

?- mytreeunique([a,b],X).
X = [a, b] .

?- mytreeunique([a,a],X).
X = [a] .

?-
```

Problem 2

```
% Name: Marco Salazar
% Date: 11/30/2020
% Assignment: Practice with Prolog Programming
% Problem: When given a binary tree as a list, I need to output the longest path from the root to a leaf.
% eg. mydepth([a,[b,[a,[c,d]]]],X).
% Yields X=8.

mydepth([], 0).
mydepth(X, 0) :- atomic(X).
mydepth([H | T], V) :- mydepth(H, V1), mydepth(T, V2), V3 is max(V1, V2), V is V3 + 1.
```

```
?- mydepth([a,[b,[a,[c,d]]]],X).
X = 8 .

?- mydepth([b,[a,[c,d]]],X).
X = 6 .

?- mydepth([a,[c,d]],X).
X = 4 .

?- mydepth([c,d],X).
X = 2 .

?- mydepth([c],X).
X = 1 .

?-
```