ECS 140A Homework 6 – Problem 3

1 Prolog

Step 3: Working Code

```
my_append([], L2, L2).
my_append([H1 | T1], L2, [H1 | T3]) :- my_append(T1 , L2 , T3).
prefix(L1, L2) :- my_append(L1, _, L2).

common_prefix(X, [H | Ys]) :- prefix(X, H), common_prefix(X, Ys).
```

Step 4: Debug Process

Bug 1

```
?- common_prefix([], [[1, 2, 3], [1, 2, 5]]).
false.
```

I did not account for the case when the list is empty which results in false, so I added a base case like rule.

Fixed code:

```
my_append([], L2, L2).
my_append([H1 | T1], L2, [H1 | T3]) :- my_append(T1 , L2 , T3).
prefix(L1, L2) :- my_append(L1, _, L2).

common_prefix(X, [H | Ys]) :- prefix(X, H), common_prefix(X, Ys).
common_prefix(X, [H | []]) :- prefix(X, H).
```

Results of given test cases:

```
?- common_prefix([], [[1, 2, 3], [1, 2, 5]]).
true .

?- common_prefix([1], [[1, 2, 3], [1, 2, 5]]).
true .

?- common_prefix([1, 2], [[1, 2, 3], [1, 2, 5]]).
true .

?- common_prefix(X, [[1, 2, 3], [1, 2, 5]]).
X = [] ;
X = [1] ;
X = [1, 2] .
```

Results of additional test cases:

```
?- common_prefix([], []).
false.

?- common_prefix([], [[]]).
true.

?- common_prefix([1, 2, 3], []).
false.

?- common_prefix([1, 2, 3], [[]]).
false.

?- common_prefix([1, 2, 3], [[1, 2]]).
false.

?- common_prefix([2, 3], [[1, 2, 3]]).
false.

?- common_prefix([5, 10], [[5, 10, 15], [5, 10], [5, 10, 15, 20]]).
true .

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

?- common_prefix(X, [[1], [2, 3], [5]]).
X = [] .

?- common_prefix(X, [[5, 10, 15, 20], [5, 10, 15], [5, 10, 15, 20, 12345]]).
X = [] ;
X = [5] ;
X = [5, 10] ;
X = [5, 10, 15] .
```

Note that the first additional test case is an interesting one. In my implementation, I consider the correct result as `false` since there's no list to compare the prefix against. However, if we want to consider the correct result as `true`, we can add another base case: `common_prefix(X, []) :- prefix(X, []).`

Step 5: Add Documentation

```
% given code from lecture
my_append([], L2, L2).
my_append([H1 | T1], L2, [H1 | T3]) :- my_append(T1 , L2 , T3).
prefix(L1, L2) :- my_append(L1, _, L2).

common_prefix(X, [H | Ys]) :- prefix(X, H), common_prefix(X, Ys). % check if X is prefix of
    first list and repeat for rest of the lists
common_prefix(X, [H | []]) :- prefix(X, H). % base case if only have 1 list to compare
    against left
```

Step 6: Extra Test Cases Used

- `common_prefix([], [])`: false
- `common_prefix([], [[]])`: true
- `common_prefix([1, 2, 3], [])`: false
- `common_prefix([1, 2, 3], [[]])`: false
- `common_prefix([1, 2, 3], [[1, 2]])`: false

- `common_prefix([2, 3], [[1, 2, 3]])`: false
- `common_prefix([5, 10], [[5, 10, 15], [5, 10], [5, 10, 15, 20]])`: true
- `common_prefix(X, [[]])`: X = []
- `common_prefix(X, [[1], [2, 3], [5]])`: X = []
- `common_prefix(X, [[5, 10, 15, 20], [5, 10, 15], [5, 10, 15, 20, 12345]])`: X = []; X = [5]; X = [5, 10]; X = [5, 10, 15]