```
/*
--------------------------------------------------------------------
PROGRAM NAME: Lab 2, Prolog Recursion
PROGRAMMER:   Samuel Jentsch
CLASS:        CSC 333, Spring 2014
INSTRUCTOR:   Dr. Strader
DATE STARTED: February 11, 2014
DUE DATE:     February 13, 2014
REFERENCES:   Dr. Strader: assignment information sheet
                                Dr. Strader: Prolog class handout.
--------------------------------------------------------------------
*/

/*********Problem 1****************************************************/

%From class Prolog handout
conc([],L,L).
conc([X|L1],L2,[X|L3]) :- conc(L1,L2,L3).


/*Remove the head from a list and append it to the end of the list.
This results in the list being "rotated" right once.*/
%Base case- An empty list is already rotated.
shift([], []).
%Base case- A list with one item is already rotated.
shift([X], [X]).
%Remove the head from the list and append it to the end.
shift([H | T], List2) :- conc(T, [H], List2).

/*********Problem 2****************************************************/

%Base case- subset of an empty set is an empty set.
subset([], []).
%Find add the head of the tail passed to the subset option.
subset([E|Tail], [E|NTail]) :- subset(Tail, NTail).
%Find the subsets in the subset tail.
subset([_|Tail], NTail) :- subset(Tail, NTail).

/*********Problem 3****************************************************/

%Calls flatten with three parameters.
flatten(List, FlatList):- flatten(List, [], FlatList).

%An empty list is already flattened.
flatten([], Flattened, Flattened).
%Flatten the head of the list and the tail of the list. Handles lists
%within the list.
flatten([H|T], L, FlatList):- flatten(H, L1, FlatList), flatten(T, L, L1).
%Adds an item that is not a list to the flattened list.
flatten(X, FlatList, [X|FlatList]):- \+ is_list(X).

/*********Problem 4****************************************************/

sumlist([], 0).
sumlist([H|T], Sum) :- sumlist(T, TailSum), Sum is H + TailSum.

/****************************************************************************/
```

Output/Verification

Problem 1:

```
?- shift([1,2,3,4,5],L1),shift(L1,L2).
L1 = [2, 3, 4, 5, 1],
L2 = [3, 4, 5, 1, 2].
```

Problem 2:

```
?- subset([a,b,c],S).
S = [a, b, c]
S = [a, b]
S = [a, c]
S = [a]
S = [b, c]
S = [b]
S = [c]
S = [].
```

Problem 3:

```
?- flatten([a,b,[c,d],[],[[[e]]],f], L).
L = [a, b, c, d, e, f]
```

Problem 4:

```
?- sumlist([1,2,3], S).
S = 6.
```