% Q1

(i) X is a list, and it is a list of lists. [0,[0,[0...]]]

(ii) Y is a list, and it is a list of 0s. [0,0,0...]

membeAcc(\_,[],Acc,Acc).

membeAcc(H,[H|T],Acc,N) :- NewAcc is Acc + 1, !, membeAcc(H,T,NewAcc,N).

membeAcc(H,[\_|T],Acc,N) :- membeAcc(H,T,Acc,N).

membe(I,L,N) :- membeAcc(I,L,0,N).

1. False.

*sublist*(L,[1,2,3,4]).

**L** = []

**L** = [1]

**L** = [1, 2]

**L** = [1, 2, 3]

**L** = [1, 2, 3, 4]

**L** = []

**L** = [2]

**L** = [2, 3]

**L** = [2, 3, 4]

**L** = []

**L** = [3]

*sublis2*(L,[1,2,3,4]).

**L** = []

**L** = [1]

**L** = []

**L** = [1, 2]

**L** = [2]

**L** = []

**L** = [1, 2, 3]

**L** = [2, 3]

**L** = [3]

**L** = []

**L** = [1, 2, 3, 4]

1. False.

*sublist*(L,[1,2,3,4]).

**L** = []

**L** = [1]

**L** = [1, 2]

**L** = [1, 2, 3]

**L** = [1, 2, 3, 4]

**L** = []

**L** = [2]

**L** = [2, 3]

**L** = [2, 3, 4]

**L** = []

**L** = [3]

*sublis3*(L,[1,2,3,4]).

**L** = []

**L** = []

**L** = []

**L** = []

**L** = []

**L** = [1]

**L** = [2]

**L** = [3]

**L** = [4]

**L** = [1, 2]

**L** = [2, 3]

(iii)

sub([],\_).

sub([X|Xs],[X|Ys]) :- sub(Xs,Ys).

sub([X|Xs],[\_|Ys]) :- sub([X|Xs], Ys).

1. True.

In findall(X,q(X),L), length(L,N),

findall can find all of the X that satisfies q(X) and put them into list L, if there are N solutions to the query q(X), then the length of the list L will be N, thus length(L,N) holds as well.

An example:

q([a]).

q([b]).

*findall*(X,*q*(X),L), *length*(L,N).

**L** = [[a], [b]],  
**N** = 2

% Q3

(a)

diff(L,L,[]).

diff(L,B,[H|T]) :- select(H,L,L1), diff(L1,B,T).

(b)

(c)

t(N) --> zeros, ones(N).

zeros --> [].

zeros --> [0], zeros.

ones(\_) --> [].

ones(N) --> {N1 is N - 1}, [1], ones(N1).

Zeros can work.

(d)

q0 --> [a], q1.

q1 --> [b], q2.

q1 --> [b], q1.

q2 --> [].