**Q1.**

list\_subset([],[]).

list\_subset([Head | Tail], [Head | Subset]):-

list\_subset(Tail, Subset).

list\_subset([Head | Tail], Subset):-

list\_subset(Tail, Subset).

**Q2.**

palindrome(List):-list\_reverse(List, List).

list\_reverse([],[]).

list\_reverse([First | Rest], Revrsed):-

list\_reverse(Rest, ReversedRest), concatenation(ReversedRest, [First], Reversed).

concatenation([], L, L).

concatenation([X1 | L1], L2, [X1 | L3]):-

concatenation(L1, L2, L3).

**Q4.**

mergesort([],[]).

mergesort([A],[A]).

mergesort([A,B | R],S):-

split([A,B | R], L1, L2), mergesort([L1,S1]), mergesort(L2, S2), merge(S1, S2, S).

split([],[],[]).

split([A],[A],[]).

split([A,B | R],[A | Ra],[B | Rb]):-

split(R, Ra, Rb).

merge(A, [], A).

merge([], B, B).

merge([A | Ra], [B | Rb], [A | M]):-

A =< B, merge[Ra, [B | Rb], M).

merge([A | Ra], [B | Rb], [B | M]):-

A>B, merge([A | Ra], Rb, M).

**Q5.**

List\_concat([], L, L)

List\_concat([X1 | L1], L2, [X1 | L3]:-

List\_concat(L1, L2, L3).

List\_rev([], []).

List\_rev([Head | Tail], Reversed):-

List\_rev(Tail, RevTail), list\_concat(RevTail, [Head], Reversed).

**Set 4 Q4.**

obs(tree).

obs(human).

obs(car).

obs(roadblock).

obs(redlight).

obs(blackbuck).

pc(noentry).

brakes(X) :-obs(X).

turnleft(X):-pc(X).

acc(X):-!,obs(X).