**Problem Statement:**

1.Define the relation:

reverse(List,ReversedList)

that reverses List.Ex:-reverse([a,b,c,d],[d,c,b,a]).

2.Define the predicate palindrome(List)

3.Define the relation translate(List1,List2) to translate a list of numbers between 0 and 9 to a list of corresponding words.Ex:-

?-translate([3,5,1,3],x).

produces: x=[three, five, one, three]

Use the following as an auxilliary relation means(0,zero),means (1,one)...means(9,nine).

4.Define the relation subset(set,subset), where set and subsets are two lists representation two sets and such that subset is the subset of set. Use it to check for subset relation as well as generate all possible subsets from a given set.

**Code:**

1.

concat([],L2,L2).

concat([H|T],L2,[H|L3]):-

concat(T,L2,L3).

reverse([],[]).

reverse([H|T],R):-

reverse(T,TR),

concat(TR,[H],R).

2.

concat([],L2,L2).

concat([H|T],L2,[H|L3]):-

concat(T,L2,L3).

reverse([],[]).

reverse([H|T],R):-

reverse(T,TR),

concat(TR,[H],R).

palindrome(L):-

reverse(L,L).

3.

means(0,zero).

means(1,one).

means(2,two).

means(3,three).

means(4,four).

means(5,five).

means(6,six).

means(7,seven).

means(8,eight).

means(9,nine).

translate([],[]).

translate([H|T],[B|E]):-

means(H,B),

translate(T,E).

4.

subset([],[]).

subset([H|T],[H|S]):-

subset(T,S).

subset([\_|T],S):-

subset(T,S).

**Results:**

1.

?-reverse([1,2,3,4],X).

X=[4,3,2,1]

2.

?-palindrome([1,2,1]).

true

?-palindrome([1,2,3]).

false

3.

?-translate([5,4,7,1],X).

X=[five,four,seven,one]

4.

?- subset([1,2,3],X).

X = [1, 2, 3] ;

X = [1, 2] ;

X = [1, 3] ;

X = [1] ;

X = [2, 3] ;

X = [2] ;

X = [3] ;

X = [].