TERMWORK - 3.

PROBLEM STATEMENT

Write a PROLOG to perform Intersection of two lists and Union of two lists.

THEORY

INTERSECTION - Intersection will return those elements that are present in both lists.
let us define a clause called list_intersection(L1,L2,L3), So this will take L1 and L2, and perform intersection operation, and store the result into L3.
So, consider L1 = [a,b,c,d,e] , L2 = [a,e,i,o,u], then L3 = [a,e].
Here, we will use the list_member() clause to check if one element is present in a list or not.

UNION - Let us define a clause called list_union(L1,L2,L3), so this will take L1 and L2, and perform Union on them, and store the result into L3.
As you know if two lists have the same element twice then after union, there will be only one. So, we need another helper clause to check the membership.
Eg: L1 = [a,b,c,d,e] , L2 = [a,e,i,o,u].
L3 = [b,c,d,e,a,i,o,u].

PROGRAM

```
intersection([X|Y],M,[X|Z]):- list_member(X,M),
                              intersection(Y,M,Z).
intersection([X|Y],M,Z):- \+ list_member(X,M),
                            intersection(Y,M,Z).

intersection([],M,[]).
list_member(X,[X|_]).
list_member(X,[_|TAIL]):- list_member(X,TAIL).


Union([X|Y],Z,W):- list_member(X,Z), union(Y,Z,W).
union([X|Y],Z,[X|W]):- \+ list_member(X,Z), union(Y,Z,W).
union([],Z,Z).
list_member(X,[X|_]).
list_member(X,[_|TAIL]):- list_member(X,TAIL).
```

OUTPUT

```
?- intersection([1,2,3,4],[1,a,b,4],X).
X = [1,4]

?- intersection([1,2,3,4],[1,a,b,5],X).
X = [1]

?- intersection([1,2,3,4],[9,a,b,5],X).
X = []

?- union([1,2,3,4],[1,a,b,4],A).
A = [2,3,1,a,b,4]

?- union([1,2,3,4],[a,b],A).
A = [1,2,3,4,a,b]

? union([1,2,3,4],[],A).
A = [1,2,3,4].

? union([1,2,3,4],[1],A).
A = [2,3,4,1].
```

CONCLUSION

In this termwork, we have learnt about how to perform intersection and union of a list.