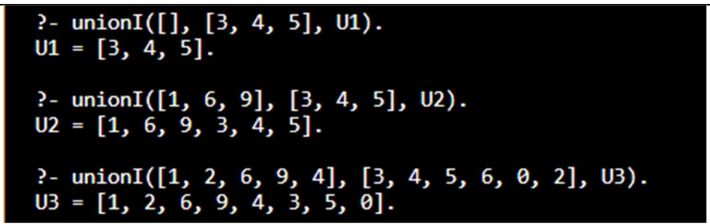
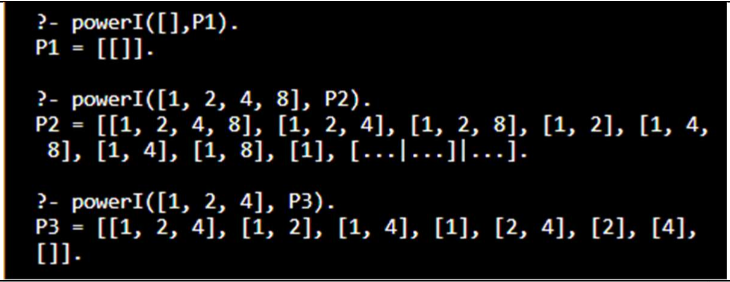
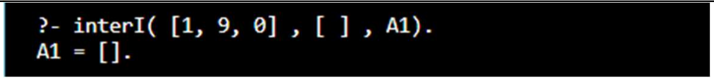
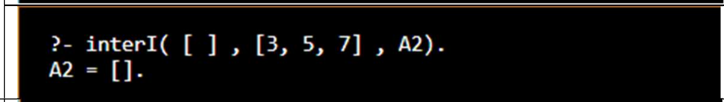
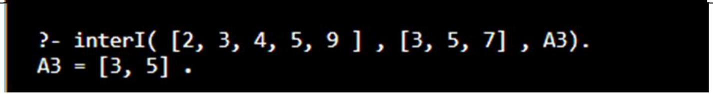
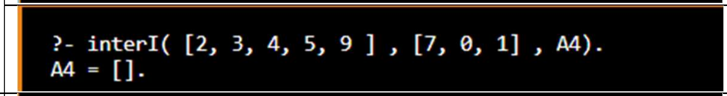
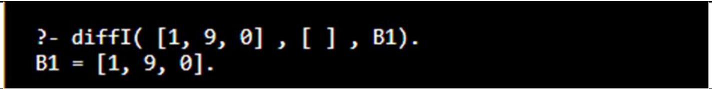
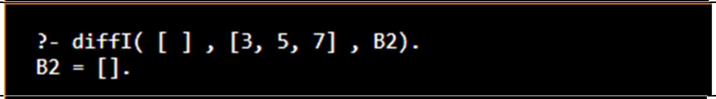
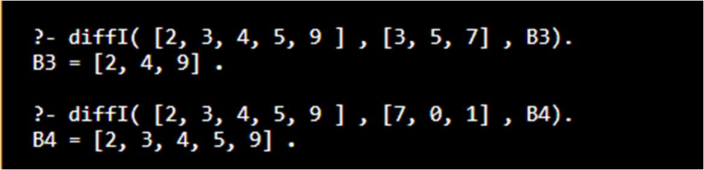
```prolog
/* membership of X in S --- from class*/
mem(X,[]) :- fail.
mem(X,[X|_]) :- !.
mem(X,[_|R]) :- mem(X,R).

% intersection
interI(S1, [ ], [ ]) :- !.
interI([ ], S2, [ ]) :- !.
interI([X|R], S2, [X|S3]) :- mem(X, S2), interI(R, S2, S3). %if X is a member of Z, put it in S3.
interI([X|R], S2, S3) :- \+mem(X, S2), interI(R, S2, S3). %if X is not a member of Z, don't put it in S3.

% difference -- opposite conditions as intersection
diffI(S1, [], S1) :- !.
diffI([], S2, []) :- !.
diffI([X|R], S2, [X|S3]) :- \+mem(X, S2), diffI(R, S2, S3).
diffI([X|R], S2, S3):- mem(X, S2), diffI(R, S2, S3).

/* append(L1, L2, L3) -- append lis  L1 to list L2 to get list  L3 -- given as reference*/
append( [ ], L, L).
append( [X|R], L, [X|Z]) :- append(R, L, Z).

/* mapcons(X,L1, L2) --  cons the element X to each list in L1 to get L2 -- given reference*/
mapcons(X, [ ], [ ]) :- !.
mapcons(X, [Y|R], [ [X|Y] | Z ]) :- mapcons(X, R, Z).

% cartesian product
cartesianI(S1, [], []) :- !.
cartesianI([], S2, []) :- !.
cartesianI([X|L], S2, S3) :- mapcons(X, S2, S2_), cartesianI(L, S2, V), append(S2_, V, S3).
```

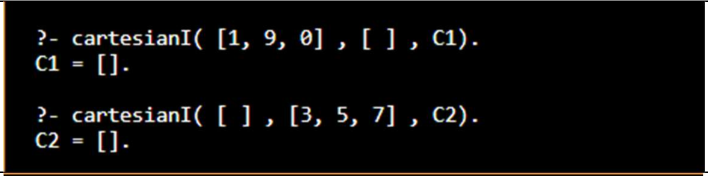
<table>
<tr><td>Checking implementation of union. The last case shows there are no duplicates.</td><td>
?- unionI([], [3, 4, 5], U1).
U1 = [3, 4, 5].

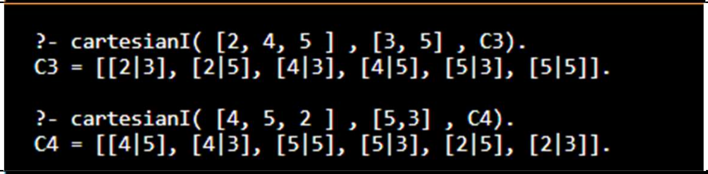
?- unionI([1, 6, 9], [3, 4, 5], U2).
U2 = [1, 6, 9, 3, 4, 5].

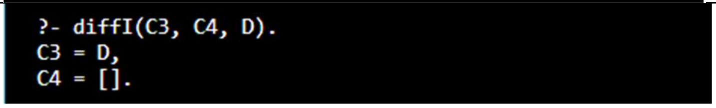
?- unionI([1, 2, 6, 9, 4], [3, 4, 5, 6, 0, 2], U3).
U3 = [1, 2, 6, 9, 4, 3, 5, 0].</td></tr>
<tr><td>Power set.</td><td>
?- powerI([],P1).
P1 = [[]].

?- powerI([1, 2, 4, 8], P2).
P2 = [[1, 2, 4, 8], [1, 2, 4], [1, 2, 8], [1, 2], [1, 4,
 8], [1, 4], [1, 8], [1], [...|...]|...].

?- powerI([1, 2, 4], P3).
P3 = [[1, 2, 4], [1, 2], [1, 4], [1], [2, 4], [2], [4],
[]].</td></tr>
<tr><td>Intersection with empty sets.</td><td>
?- interI([1, 9, 0] , [ ] , A1).
A1 = [].

?- interI([ ] , [3, 5, 7] , A2).
A2 = [].</td></tr>
<tr><td>Intersection examples.</td><td>
?- interI([2, 3, 4, 5, 9] , [3, 5, 7] , A3).
A3 = [3, 5] .

?- interI([2, 3, 4, 5, 9] , [7, 0, 1] , A4).
A4 = [].</td></tr>
<tr><td>Set difference examples.</td><td>
?- diffI([1, 9, 0] , [ ] , B1).
B1 = [1, 9, 0].

?- diffI([ ] , [3, 5, 7] , B2).
B2 = [].

?- diffI([2, 3, 4, 5, 9] , [3, 5, 7] , B3).
B3 = [2, 4, 9] .

?- diffI([2, 3, 4, 5, 9] , [7, 0, 1] , B4).
B4 = [2, 3, 4, 5, 9] .</td></tr>
<tr><td>Cartesian product with empty sets</td><td>
?- cartesianI([1, 9, 0] , [ ] , C1).
C1 = [].

?- cartesianI([ ] , [3, 5, 7] , C2).
C2 = [].</td></tr>
<tr><td>Cartesian product with same sets but different orders.</td><td>
?- cartesianI([2, 4, 5] , [3, 5] , C3).
C3 = [[2|3], [2|5], [4|3], [4|5], [5|3], [5|5]].

?- cartesianI([4, 5, 2] , [5,3] , C4).
C4 = [[4|5], [4|3], [5|5], [5|3], [2|5], [2|3]].</td></tr>
<tr><td>To check that order doesn’t matter, take difference of both products. It results in an empty set.</td><td>
?- diffI(C3, C4, D).
C3 = D,
C4 = [].</td></tr>
</table>