Practical – 6

Implement following prolog programs based on list.

a. To copy one list to another list.
b. To append one list to another list.
c. To find the intersection of two lists.
d. To find the union of two lists.

A: To copy one list into another list

```
copy_list([], []).

copy_list([H|T1], [H|T2]):-
        copy_list(T1, T2).
```

Output :

```
?-
% e:/AHD/SEM 8/AI/p6a.pl compiled 0.00 sec, 2 clauses
?- copy_list([2,4,6,1,3],A).
A = [2, 4, 6, 1, 3].

?-
```

B: To append one list to another list

```
append_lists([], [], []).

append_lists([], [H2|T2], [H2|T3]):-
        append_lists([], T2, T3).

append_lists([H1|T1], L2, [H1|T3]):-
        append_lists(T1, L2, T3).
```

Output :

```
?-
% e:/AHD/SEM 8/AI/p6a.pl compiled 0.00 sec, 0 clauses
?- append_lists([2,4],[3,1,7],A).
A = [2, 4, 3, 1, 7]
```

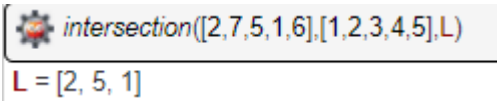
C:To find the intersection of two lists.

```
intersection([], _, []).

intersection([H|T1], L2, [H|T3]):-
        member(H, L2),
        intersection(T1, L2, T3).

intersection([_|T1], L2, L3):-
        intersection(T1, L2, L3).
```

Output :



D:To find the union of two lists.

```
union([], L2, L2).

union([H1|T1], L2, [H1|T3]):-
        not(member(H1, L2)),
        union(T1, L2, T3).

union([_|T1], L2, L3):-
        union(T1, L2, L3).
```

Output :