ASSIGNMENT I

1. Write a prolog program to compute the sum of the list.
   Ans.

```
sum([],0).
sum([X|T],Sum):-
   sum(T,Sum1),
   Sum is Sum1 + X.
```

2. Write a prolog program to find the maximum of two elements.
   Ans.

```
max(X,Y,X):-
   X>=Y.
max(X,Y,Y):- X<Y.
```

3. Write a prolog program to find the length of the list.
   Ans.

```
len([], 0).
len([_|T], N):-
   len(T, N1),
   N is N1 + 1.
```

4. Write a prolog program to find the GCD.
   Ans.

```
gcd(X,Y,Y):-
   K is mod(X,Y),
   K == 0.
gcd(X,Y,Z):-
   Y>X,
   gcd(Y,X,Z).
gcd(X,Y,Z):-
   K is mod(X,Y),
        K \= 0,
        gcd(Y,K,Z).
```

5. Write a prolog program to check same length.
   Ans.

```
len_same([],[]):- writeln("Yes, Same Length"), !.
len_same([],_):- writeln("Not Same Length"), !.
len_same(_,[]):- writeln("Not Same Length"), !.

len_same([_|T1],[_|T2]):-
```

```
    len_same(T1,T2).
```

6. Write a prolog program to concatenate of two list.
   Ans.

```
concat([],L,L).
concat([H|T], L, [H|R]):-
    concat(T,L,R).
```

7. Write a prolog program to find out the maximum element of list.
   Ans.

```
max_list([],0).
max_list([H|T],X):-
    max_list(T,X1),
    H >= X1,
    X is H.
max_list([H|T],X):-
    max_list(T,X1),
    H < X1,
    X is X1.
```

8. Write a prolog program to find out the factorial of an element.
   Ans.

```
fact(0,1).
fact(1,1).
fact(N,F):-
    N1 is N - 1,
    fact(N1, F1),
    F is N*F1.
```

ASSIGNMENT II

1. Write a prolog program to find fibonacci series.

Ans.

```
fibonacci(1,[0]).
fibonacci(2,[1,0]).

fibonacci(N,[R,X,Y | Tail]) :-
   N > 2,
   N1 is N - 1,
   fibonacci(N1, [X,Y|Tail]),
   R is X + Y.

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


fib(N,Y):-
   fibonacci(N,X),
   reverse(X,Y,[]).
```

2. Write a prolog program to test whether a list is a double header or not.

Ans.

```
doubleHeaded([],0) :- write("No").
doubleHeaded([],1) :- write("No").
doubleHeaded([H|T],2) :- write("Yes").
doubleHeaded([], N) :-
   N>=2,
   write("Yes").
doubleHeaded([H|T], N) :-
   N1 is N+1,
   doubleHeaded(T, N1).
```

3. Write a prolog program to test whether a list is not exactly of two elements list.

   Ans.

```
twoTest(2,[]):- write("Exactly 2 elements").
twoTest(2, [H|T]) :- write("Not Exactly 2 elements").
```

```
twoTest(N, []) :-
   N<2,
   write("Not Exactly 2 elements").

twoTest(N, [H|T]) :-
   N<2,
   N1 is N + 1,
   twoTest(N1, T).
```

4. Write a prolog program to determine whether an element X is a member of a list L.
   Ans.

```
search(X, []) :- write("Not in List").
search(X, [H|T]) :-
   X == H,
   write("In List").
search(X, [H|T]):-
   X \= H,
   search(X, T).
```

5. Write a prolog program to find the reverse of a list.
   Ans.

```
revL([],X,X).
revL([H|T],X,L) :- revL(T,X,[H|L]).
```

6. Write a prolog program to add an element.

Ans.

```
list_insert(D, L, X):-
   list_delete(D, X, L).

list_delete(D, [D|List1], List1).
list_delete(D, [Y|List1], [Y|List2]) :-
   list_delete(D, List1, List2).
```

7. Write a prolog program to define a predicate between which generates all integers X.
   Ans.

```
allInt(X) :- allInt(1,X).

allInt(X,X) :- write(X).
allInt(X,Y):- X>Y.
```

```
allInt(X,Y):-
   X<Y,
   write(X),
   write(", "),
   X1 is X + 1,
   allInt(X1, Y).
```

8. Write a prolog program to find last element.
   Ans.

```
last(X,[X]):- write("Yes").
last(X,[]):- write("No").
last(X, [_|Tail]):- last(X,Tail).
```

ASSIGNMENT III

1. Write a prolog program to delete all occurrence of an element.
   Ans.

```
del(_,[],[]).
del(X,[X|Tail],R):-
   del(X,Tail,R).
del(X,[Y|Tail],[Y|R]):-
   X\=Y,
   del(X,Tail,R).
```

2. Write a prolog program to test whether a list X is a subset of a list Y.
   Ans.

```
subset([],[]):- writeln("Subset").
subset([],_):- writeln("Subset").
subset([H|Tail],T):-
   memberchk(H,T),
   subset(Tail,T).
subset([H|_],T):-
   \+ memberchk(H,T),
        !, writeln("Not Subset").
```

3. Write a prolog program to intersect of two list X and Y.
   Ans.

```
intersect([],_,[]).
intersect([H|Tail],T,R):-
   memberchk(H,T),
   !,
   intersect(Tail,T, RTail),
   R = [H|RTail].


intersect([H|Tail],T,R):-
   \+ memberchk(H,T),
        intersect(Tail,T, R).
```

4. Write a prolog program to union of two list X and Y,
   Ans.

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

```
union([H|Tail],T,R):-
   memberchk(H,T),!,
   union(Tail,T,R).

union([H|Tail],T,R):-
   \+ memberchk(H,T),
         !,
   union(Tail,T, RTail),
   R = [H|RTail].

unionset(X,Y,R):-
   union(X,Y,Z),
   append(Y,Z,R).
```

5. Write a prolog program to divide a list in two list which are appropriately of same length.

Ans.

```
left(0,X,One,Two):-
   Two = X,
   One = [].
left(L,[H|Tail],One,Two):-
   L1 is L - 1,
   left(L1,Tail,OneTail,Two),
   One = [H|OneTail].

split(X,One,Two):-
   length(X,L),
         L1 is div(L,2),
   left(L1,X,One,Two).
```

ASSIGNMENT IV

1. Write a prolog program to find the maximum of two elements using CUT.
   Ans.

```
max(X,Y,X):-
   X>=Y, !.
max(X,Y,Y):- X<Y.
```

2. Write a prolog program to sum of a list using accumulator.
   Ans.

```
sum([],Acc,Acc).
sum([H|T],Acc, Sum):-
   Sum1 is H + Acc,
   sum(T,Sum1,Sum).
```

3. Write a prolog program to length of list using accumulator.
   Ans.

```
length([],Acc,Acc).
length([_|T],Acc,N):-
   Ac is Acc + 1,
   length(T,Ac,N).
```

4. Write a prolog program to find the maximum of a list elements using CUT.
   Ans.

```
max([],Acc,Acc).
max([H|T],Acc,N):-
   H > Acc,
   Ac = H, !,
   max(T,Ac,N).
max([H|T],Acc,N):-
   H =< Acc, !,
   max(T,Acc,N).
```

5. Write a prolog program to GCD of two elements with CUT.
   Ans.

```
gcd(X,X,X):- !.
gcd(X,Y,Y):-
   K is mod(X,Y),
   K == 0, !.
gcd(X,Y,Z):-
   K is mod(X,Y),
```

```
    K \= 0, !,
    gcd(Y,K,Z).
```

6. Write a prolog program to find the GCD of list.

   Ans.

```
gcd(X,X,X):- !.
gcd(X,Y,Z):-
    Y > X,
    gcd(Y,X,Z).
gcd(X,Y,Y):-
    K is mod(X,Y),
    K == 0, !.
gcd(X,Y,Z):-
    K is mod(X,Y),
    K \= 0, !,
    gcd(Y,K,Z).
gcd_rec([],Temp,Temp).
gcd_rec([H|T], Temp, X):-
    gcd(H,Temp,Z),
    gcd_rec(T,Z,X).
gcd_list([H1, H2| T], X):-
    gcd(H1,H2,Z),
    gcd_rec(T,Z,X).
```

7. Write a prolog program to reverse of list using accumulator.

   Ans.

```
rev(L,X):-
    reva(L,[],X).
reva([],Acc,Acc).
reva([H|T], Acc, X):-
    Ac = [ H | Acc ],
    reva(T, Ac, X).
```

ASSIGNMENT V

1. Write a prolog program to select an element from a list.
2. Write a prolog program to sort all the elements of a list using merge sort.
   Ans.

```
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|R1],[B|R2]):-
   split(R,R1,R2).

merge([],[],[]).
merge(A,[],A).
merge([],A,A).
merge([A|T1],[B|T2],[A|S1]):-
   A =< B,
   merge(T1, [B|T2], S1).
merge([A|T1],[B|T2],[B|S1]):-
   B < A,
   merge([A|T1], T2, S1).
```

3. Write a prolog program to sort all the elements of a list using quick sort.
   Ans.

```
quicksort([],[]).
quicksort([A],[A]).
quicksort([A|R],S):-
        partition(A,R,L1,L2),
   quicksort(L1,S1),
   quicksort(L2,S2),
   append(S1,[A|S2],S).

partition(_,[],[],[]).
partition(A, [T|R], [T|R1], L2):-
   A >= T, !,
```

```
    partition(A,R,R1,L2).
partition(A, [T|R], L1, [T|R2]):-
    A < T, !,
    partition(A,R,L1,R2).
```

4. Write a prolog program to sort all the elements of a list using permutation sort.
   Ans.

```
permutationsort(List,Sorted):-
    permutation(List,Sorted),
    is_sorted(Sorted).

is_sorted([]).
is_sorted([_]).
is_sorted([X,Y|T]):-X=<Y,is_sorted([Y|T]).
```

5. Write a prolog program to sort all the elements of a list using insertion sort.
   Ans.

```
inssort([],[]).
inssort([A],[A]).
inssort([A|R],S):-
        inssort(R,S1),
    insertion(A,S1,S).

insertion(A,[],[A]).
insertion(A,[H|T],[A,H|T]):-
    A =< H, !.
insertion(A,[H|T],[H|S1]):-
    A > H, !,
    insertion(A,T,S1).
```

6. Write a prolog program to sort all the elements of a list using selection sort.
   Ans.

```
selectionsort([],[]).
selectionsort([First|Rest], [Smallest|SortedList]) :-
        smallest(Rest, First, Smallest),
        remove([First|Rest], Smallest, NewList),
        selectionsort(NewList, SortedList),!.

smallest([], Smallest,Smallest).
smallest([First|Rest], CurrSmallest, Smallest) :-
```

```
        First < CurrSmallest, smallest(Rest, First, Smallest).
smallest([_|Rest], CurrSmallest, Smallest) :-
        smallest(Rest, CurrSmallest, Smallest).

remove([], _, []).
remove([First|Rest], First, Rest).
remove([First|Rest], Element, [First|NewList]) :-
                remove(Rest, Element, NewList).
```

7. Write a prolog program to sort all the elements of a list using bubble sort.

Ans.

```
bubbleSort(L,SL):-
   swap(L,L1),!,
   bubbleSort(L1,SL).
bubbleSort(L,L).
swap([X,Y|Rest],[Y,X|Rest]):-
   X > Y, ! .
swap([Z|Rest],[Z|Rest1]):-
   swap(Rest,Rest1).
```