**1.Greatest common divisor**

gcd(NR1, NR2, R):-

NR1 = NR2,

R = NR1.

gcd(NR1, NR2, R):-

NR1 < NR2,

NR is NR2-NR1,

gcd(NR1, NR, R).

gcd(NR1, NR2, R):-

NR1 > NR2,

gcd(NR2, NR1, R).

**2.Insert to position (can add conditions for last position and so on)**

insertPos([], \_, \_, \_,[]).

insertPos([H|T], E, POS, INDEX, [H|R]):-

POS =\= INDEX,

INDEX1 = INDEX +1,

insertPos(T, E, POS, INDEX1, R).

insertPos([H|T], E, POS, INDEX, [E|R]):-

POS =:= INDEX,

insertPos([H|T], E, 0, INDEX, R).

insert(L, E, POS, R):-

insertPos(L, E, POS, 1, R).

**3.Remove ALL occurrences of atom**

removeOcc([], \_, []).

removeOcc([H|T], E, R):-

H =:= E,

removeOcc(T, E, R).

removeOcc([H|T], E, [H|R]):-

H =\= E,

removeOcc(T, E, R).

**4. Remove first k(or next depends of the params!!) occurrences**

rem3([], \_, \_,[]).

rem3([H|T], E, K, R):-

H =:= E,

K > 0,

K1 is K - 1,

rem3(T, E, K1, R).

rem3([H|T], E, K, [H|R]):-

rem3(T, E, K,R).

**5.Count occurrences**

countOcc([], \_, C, R):-

R is C.

countOcc([H|T], E, C, R):-

H =:= E,

C1 is C+1,

countOcc(T, E, C1, R).

countOcc([H|T], E, C, R):-

H =\= E,

countOcc(T, E, C, R).

**6.Find max**

findMax([], E, MAX):-

MAX is E.

findMax([H|T], E,MAX):-

H > E,

findMax(T, H, MAX).

findMax([H|T], E,MAX):-

H =< E,

findMax(T, E, MAX).

**7.Diffrence of 2 sets**

diffSet(L1, [], R):-

R = L1.

diffSet([], \_, R):-

R = [].

diffSet([H1|T1], [H2|T2], R):-

removeOcc([H1|T1], H2, R1),

diffSet(R1, T2, R).

**8.Merge lists**

mergeList([],L,L ).

mergeList([H|T],L,[H|M]):-

mergeList(T,L,M).

**9.Search**

search([], \_, R):-

R is 1.

search([H|\_], E, R):-

H =:= E,

R is 0,

!.

search([H|T], E, R):-

H =\= E,

search(T, E, R).

**10.Check if a set**

isSet([], R):-

R is 0.

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

search(T, H, FOUND),

FOUND =:= 0,

R is 1,

!.

isSet([\_|T], R):-

isSet(T, R).

**11.Split a list**

split([],[],[],0,0).

split([H|T],[H|Odd],Even,O,E):-

H mod 2 =:= 1,

split(T, Odd, Even, O1, E),

O is O1+1.

split([H|T],Odd,[H|Even],O,E):-

H mod 2 =:= 0,

split(T, Odd, Even, O, E1),

E is E1+1.

**12.Contains element**

contains([V|\_], V) :- !.

contains([\_|T], E) :- contains(T, E).

**13.intersection of sets**

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

intersection([H|T], B, [H|R]) :-

contains(B, H),

intersection(T, B, R), !.

intersection([\_|T], B, R) :- intersection(T, B, R).

**14.Valley aspect**