**PRIME**

divisible(X,Y):-

N is Y\*Y,

N =< X,

X mod Y =:= 0.

divisible(X,Y):-

Y < X,

Y1 is Y+1,

divisible(X,Y1).

isprime(X):-

Y is 2, X > 1, \+divisible(X,Y).

isprime(7).

**MAX NO LIST**

% Base case: The maximum element of a list with a single element is that element.

max\_in\_list([X], X).

% Recursive rule: Compare the first element with the maximum of the rest of the list.

max\_in\_list([H|T], Max) :-

max\_in\_list(T, MaxRest),

Max is max(H, MaxRest).

max\_in\_list([1, 5, 3, 7, 2, 8], Max).

**PALINDROME OR NOT**

% Base case: An empty list is a palindrome.

is\_palindrome([]).

% Base case: A list with a single element is a palindrome.

is\_palindrome([\_]).

% Recursive rule: Check if the first and last elements are equal,

% and recursively check if the rest of the list is a palindrome.

is\_palindrome([H|T]) :-

append(Middle, [H], T),

is\_palindrome(Middle).

is\_palindrome([1, 2, 3, 2, 1]).

**FIRST TWO ELEMENTS OR NOT**

% Base case: An empty or single-element list does not have distinct first two elements.

first\_two\_elements\_same([]) :- false.

first\_two\_elements\_same([\_]) :- false.

% Rule: Check if the first and second elements are the same.

first\_two\_elements\_same([X, X | \_]) :- true.

% Rule: If the first and second elements are different, then they are not the same.

first\_two\_elements\_same([X, Y | \_]) :- X \= Y.

first\_two\_elements\_same([1, 1, 2, 3]).

**POSITIVE INTEGER CO PRIME**

% Base case: The GCD of any number N with 1 is 1.

gcd(N, 1, 1) :- !.

% Recursive rule to find the GCD using Euclidean algorithm.

gcd(M, N, G) :-

N > 0,

R is M mod N,

gcd(N, R, G).

% Predicate to check if two numbers are coprime.

coprime(X, Y) :-

gcd(X, Y, G),

G =:= 1.

coprime(15, 28).

**SUM OF ELEMENTS**

% Base case: The sum of an empty list is 0.

sum\_list([], 0).

% Recursive rule: Sum the first element with the sum of the rest of the list.

sum\_list([Head|Tail], Sum) :-

sum\_list(Tail, RestSum),

Sum is Head + RestSum.

sum\_list([1, 2, 3, 4, 5], Sum).

**GCD OF TWO NOS**

% Base case: The GCD of any number N with 0 is N.

gcd(N, 0, N) :- !.

gcd(0, N, N) :- !.

% Recursive rule to find the GCD using the Euclidean algorithm.

gcd(M, N, G) :-

M > 0,

R is M mod N,

gcd(N, R, G).

gcd(M, N, G) :-

M > N,

R is N mod M,

gcd(M, R, G).

gcd(48, 18, Result).