```
rev([],[]).
rev([H|T],L1):-
   rev(T,L2),
   conc(L2,[H],L1).

conc([],L,L).
conc([H|T],L,[H|L1]):-
   conc(T,L,L1).

sum([],0).
sum([H|T],X):-
   sum(T,X1),
   X is X1+H.

exist(X,[X|_]).
exist(X,[_|T]):-
   exist(X,T).

insert_at(X,L,1,[X|L]).
insert_at(X,[H|T],P,[H|R]):-
   P>1,
   K1 is P-1,
   insert_at(X,T,K1,R).

del(1,[_|T],T).
del(X,[H|T],[H|R]):-
   X>1,
   X1 is X-1,
   del(X1,T,R).

% Base case: anything to the power 0 is 1
power(_, 0, 1).

% Recursive case: Base^Exponent = Base * Base^(Exponent-1)
power(Base, Exponent, Result) :-
   Exponent > 0,
   Exponent1 is Exponent - 1,
   power(Base, Exponent1, Result1),
   Result is Base * Result1.
```