1-

```
Rules - nth(1,[T|_],T).
nth(N,[_|T],X) :- N1 is N-1,nth(N1,T,X).
?- nth(1,[a,b,c,d],X).
X = a .
?- nth(3,[a,b,c,d],X).
X = c .
```

2-

```
Rule - leng([],0).
leng([H|T],X) :- leng(T,X1),X is X1+1.
?- leng([a,b,c,d,e],X).
X = 5.
?- leng([a,d,e],X).
X = 3.


member([H|_],H).
member([_|T],H) :- member(T,H).
?- member([a,b,c,d,e],e).
true .
?- member([a,b,c],d).
false.

append([],L,L).
append([H|T],L2,[H|R]) :- append(T,L2,R).
?- append([b,c,d],[e,f,g],L).
L = [b, c, d, e, f, g].

las([X],X).
las([H|T],X) :- las(T,X).
?- last([e,f,g],X).
X = g.
?- last([h,i,j,k],X).
X = k.

rev([],L,L).
rev([H|R],F,L) :- rev(R,F,[H|L]).
?- rev([a,b,c,d],X,[]).
X = [d, c, b, a].
```

3-

```
divisible(X,Y) :- X>Y,(0 is mod(X,Y);divisible(X,Y+1)).
prime(X) :- not(divisible(X,2)).
check(N,X,E) :- (X<div(N,2)+1),((prime(X),N1 is N-X,prime(N1),E =X+N1);(X1 is
X+1,check(N,X1,E))).
goldbach(N,E) :- check(N,2,E).

?- goldbach(24,E).
E = 5+19 .
?- goldbach(86,E).
E = 3+83 .
```

4-

```
move(state(middle,onbox,middle,hasnot),get,state(middle,onbox,middle,has)).
move(state(P,onfloor,P,hasnot),climb,state(P,onbox,P,hasnot)).
move(state(P1,onfloor,P1,hasnot),push(P1,P2),state(P2,onfloor,P2,hasnot))
move(state(P,onfloor,B,hasnot),walk(P,P1),state(P1,onfloor,B,hasnot))

ispossible(state(_,_,_,has)).
ispossible(State) :- move(State,Move,State1),ispossible(State1),write(Move),nl.

walk(atdoor,atwindow)
push(atwindow,middle)
climb
get
```

5-

```
Fill 4L jug
Fill 3L jug
Empty 4L jug
Empty 3L jug into 4L jug until 4L is full
Fill 3L jug
Empty 3L jug into 4L jug until 4L is full
Empty 4L jug
Empty 3L jug into 4L jug until 4L is full
true .
```