```prolog
%%% BIBLICAL FAMILY
% father(X, Y): X is Y's father.
father(terach, abraham).
father(terach, nachor).
father(terach, haran).
father(abraham, isaac).
father(haran, lot).
father(haran, milcah).
father(haran, yiscah).

% mother(X, Y): X is Y's mother.
mother(sarah, isaac).

% male(X): X is male.
male(terach).
male(abraham).
male(nachor).
male(haran).
male(isaac).
male(lot).

% female(X): X is female.
female(sarah).
female(milcah).
female(yiscah).

% person(X): X is a person.
person(X):- male(X).
person(X):- female(X).

% parent(X, Y): X is Y's parent.
parent(X, Y):- father(X, Y).
parent(X, Y):- mother(X, Y).

% likes(X, Y): X likes Y.
/*
likes(terach, apples).
likes(abraham, apples).
likes(nachor, apples).
likes(haran, apples).
likes(isaac, apples).
likes(lot, apples).
likes(sarah, apples).
likes(milcah, apples).
likes(yiscah, apples).
```

```prolog
*/

likes(X, apples):- person(X).

% grandparent(X, Y): X is Y's grandparent.
grandparent(X, Y):- father(X, Z), father(Z, Y).
grandparent(X, Y):- father(X, Z), mother(Z, Y).
grandparent(X, Y):- mother(X, Z), father(Z, Y).
grandparent(X, Y):- mother(X, Z), mother(Z, Y).

% ancestor(X, Y): X is Y'ancestor.
% Base component
ancestor(X, Y):- parent(X, Y).
% Recursive component
ancestor(X, Y):- parent(X, Z), ancestor(Z, Y).

% sister(X, Y): X is Y's sister.
sister(X, Y):- parent(Z, X), parent(Z, Y), female(X), not(X = Y).

% brother(X, Y): X is Y's brother.
brother(X, Y):- parent(Z, X), parent(Z, Y), male(X), not(X = Y).

% sibling(X, Y): X is Y's sibling.
sibling(X, Y):- sister(X, Y).
sibling(X, Y):- brother(X, Y).
```