% ------------------------------------

% Family Tree Database

% ------------------------------------

% gender facts

male(john).

male(peter).

male(mike).

male(david).

male(kevin).

female(linda).

female(susan).

female(mary).

female(sophia).

female(anna).

% parent facts

parent(john, peter).

parent(linda, peter).

parent(john, mary).

parent(linda, mary).

parent(peter, mike).

parent(susan, mike).

parent(peter, sophia).

parent(susan, sophia).

parent(mary, david).

parent(kevin, david).

parent(mary, anna).

parent(kevin, anna).

% ------------------------------------

% Rules

% ------------------------------------

% father(X,Y) :- X is father of Y

father(X, Y) :-

male(X),

parent(X, Y).

% mother(X,Y) :- X is mother of Y

mother(X, Y) :-

female(X),

parent(X, Y).

% sibling(X,Y) :- X and Y share a parent

sibling(X, Y) :-

parent(Z, X),

parent(Z, Y),

X \= Y.

% brother(X,Y) :- X is male sibling of Y

brother(X, Y) :-

sibling(X, Y),

male(X).

% sister(X,Y) :- X is female sibling of Y

sister(X, Y) :-

sibling(X, Y),

female(X).

% grandparent(X,Y) :- X is grandparent of Y

grandparent(X, Y) :-

parent(X, Z),

parent(Z, Y).

% ------------------------------------

% Main Program (demonstration)

% ------------------------------------

:- initialization(main).

main :-

nl, write('--- Family Tree Demo ---'), nl, nl,

write('Father of Mary: '), father(F, mary), write(F), nl,

write('Mother of Mary: '), mother(M, mary), write(M), nl, nl,

write('Siblings of Mary: '), nl,

forall(sibling(mary, S), (write(' - '), write(S), nl)), nl,

write('Children of Peter: '), nl,

forall(parent(peter, C), (write(' - '), write(C), nl)), nl,

write('Grandparents of Sophia: '), nl,

forall(grandparent(G, sophia), (write(' - '), write(G), nl)), nl,

halt.