Artificial Intelligence

Asssignment 5

Name : Swati Anil Sonone

Div : CS\_D

Roll no. : 53

Problem Statement : Family Tree using prolog

Code :

% Facts

male(mahipat).

male(baban).

male(vaman).

male(motiram).

male(kunal).

male(anil).

male(dharmapal).

male(vinod).

male(sumedh).

male(pawan).

male(rahul).

male(rudra).

male(sammek).

female(gangubai).

female(ganga).

female(vaishali).

female(vishakha).

female(sneha).

female(swati).

female(anamika).

female(jija).

female(pradnya).

female(suman).

female(chitra).

female(kalpana).

parent(mahipat, motiram).

parent(mahipat,vaman).

parent(mahipat,baban).

parent(mahipat,gangubai).

parent(motiram,anil).

parent(motiram,kunal).

parent(motiram,dharmapal).

parent(kunal,vaishali).

parent(kunal,vishakha).

parent(chitra,anamika).

parent(chitra,vaishali).

parent(chitra,vishakha).

parent(kunal,anamika).

parent(anil,sneha).

parent(anil,swati).

parent(anil,pawan).

parent(dharmapal,rahul).

parent(rahul,rudra).

parent(vaishali,sammek).

parent(vaman,vinod).

parent(vinod,sumedh).

parent(vinod,pradnya).

parent(kalpana,sumedh).

parent(kalpana,pradnya).

parent(pradnya,aarohi).

parent(ganga,jija).

parent(jija,sneha).

parent(jija,swati).

parent(jija,pawan).

parent(suman,kunal).

parent(suman,anil).

parent(suman,dharmapal).

couple(anil,jija).

couple(motiram,suman).

couple(vinod,kalpana).

couple(kunal,chitra).

father(X, Y) :- male(X), parent(X, Y).

mother(X, Y) :- female(X), parent(X, Y).

brother(X,Y) :- male(X),parent(W,X),parent(W,Y).

sister(X,Y):- female(X), parent(W,X),parent(W,Y).

wife(X,Y) :- female(X),male(Y), parent(X,Z),parent(Y,Z).

husband(X,Y) :- female(Y),male(X), wife(Y,X).

father\_in\_law(X, Y) :- male(X),female(Y),parent(X,T),wife(Y,T).

sec\_father\_in\_law(X, Y) :- sibling(X,W),male(W),female(Y),parent(W,T),wife(Y,T).

mother\_in\_law(X, Y) :- female(X),female(Y),parent(Y,T),grandmother(X,T).

son(X, Y) :- male(X), parent(Y, X).

daughter(X, Y) :- female(X), parent(Y, X).

grandfather(X, Z) :- father(X, Y), parent(Y, Z).

grandchild(X,Z) :- grandfather(Z,X).

deer(X,Y) :- male(W),male(X),sibling(X,W),father(W,T),mother(Y,T).

deerani(X,Y) :- mother(X,W),cousin(W,T),mother(Y,T).

grandmother(X, Z) :- mother(X, Y), parent(Y, Z).

sibling(X, Y) :- parent(Z, X), parent(Z, Y), X \= Y.

cousin(X, Y) :- parent(Z, X), parent(W, Y), sibling(Z, W).

uncle(X, Y) :- parent(Z, Y), sibling(X, Z), male(X).

aunt(X, Y) :- cousin(Z, Y), mother(X, Z), female(X).

mat\_aunt(X,Y) :- female(X),sister(X,W),mother(W,Y).

sec\_mat\_aunt(X,Y):- female(X),cousin(X, W),mother(W,Y).

sec\_sec\_mat\_aunt(X,Y):- female(X),second\_cousin(X, W),mother(W,Y).

sec\_mama(X,Y) :- male(X),cousin(X,W),mother(W,Y).

sec\_sec\_mama(X,Y):- male(X),second\_cousin(X,W),mother(W,Y).

second\_uncle(X, Y) :-male(X),grandfather(W, Y), sibling(W,Z),son(X,Z).

second\_aunt(X, Y) :-female(X),grandfather(W, Y), sibling(W,Z),grandfather(Z,D),mother(X,D).

second\_cousin(X, Y) :- grandfather(W, Y), sibling(W,Z),grandfather(Z,X).

cousin\_sibling(X,Y) :- parent(W,X),parent(Z,Y),cousin(W,Z).

sec\_cousin\_sibling(X,Y) :- parent(W,X),parent(Z,Y),second\_cousin(W,Z).

second\_grandfather(X,Y) :- grandfather(W,Y),sibling(W,X).

second\_grandmother(X,Y) :- second\_uncle(W,Y),mother(X,W).

great\_grandfather(X, Z) :- father(X, Y), grandfather(Y, Z).

great\_grandchild(X,Z) :- great\_grandfather(Z,X).

great\_grandmother(X, Z) :- mother(X, Y), grandfather(Y, Z).

sec\_great\_grandfather(X,Y) :-great\_grandfather(W,Y),sibling(W,X).

wife\_of\_brother\_of\_husband(X,Y) :- husband(Z,X),husband(D,Y),cousin(Z,D).

brother\_in\_law(X,Y) :- male(X),brother(X,W),couple(Y,W).

sec\_sec\_grand\_mother(X,Y) :- parent(W,Y),second\_aunt(X, W).

sec\_sister\_in\_law(X,Y):-wife(X,W),cousin(W,Y).

sec\_great\_grandmother(X,Y) :-great\_grandmother(W,Y),sibling(W,X).

% Your family relationships and rules here

find\_relationship(X, Y) :-

father(X, Y), write(X), write(' is the father of '), write(Y), nl.

find\_relationship(X, Y) :-

wife(X, Y), write(X), write(' is the wife of '), write(Y), nl.

find\_relationship(X, Y) :-

mother(X, Y), write(X), write(' is the mother of '), write(Y), nl.

find\_relationship(X, Y) :-

son(X, Y), write(X), write(' is the son of '), write(Y), nl.

find\_relationship(X, Y) :-

daughter(X, Y), write(X), write(' is the daughter of '), write(Y), nl.

find\_relationship(X, Y) :-

grandmother(X, Y), write(X), write(' is the grandmother of '), write(Y), nl.

find\_relationship(X, Y) :-

grandfather(X, Y), write(X), write(' is the grandfather of '), write(Y), nl.

find\_relationship(X, Y) :-

sibling(X, Y), write(X), write(' is the sibling of '), write(Y), nl.

find\_relationship(X, Y) :-

brother(X, Y), write(X), write(' is the brother of '), write(Y), nl.

find\_relationship(X, Y) :-

sister(X, Y), write(X), write(' is the sister of '), write(Y), nl.

find\_relationship(X, Y) :-

father(Z, Y), mother(X, Z), write(X), write(' is the grandmother of '), write(Y), nl.

find\_relationship(X, Y) :-

second\_grandmother(X,Y), write(X), write(' is the second grandmother of '), write(Y), nl.

find\_relationship(X, Y) :-

second\_grandfather(X,Y), write(X), write(' is the second grandfather of '), write(Y), nl.

find\_relationship(X, Y) :-

second\_uncle(X, Y), write(X), write(' is the second-uncle of '), write(Y), nl.

find\_relationship(X, Y) :-

second\_aunt(X, Y), write(X), write(' is the second-aunt of '), write(Y), nl.

find\_relationship(X, Y) :-

uncle(X, Y), write(X), write(' is the uncle of '), write(Y), nl.

find\_relationship(X, Y) :-

sec\_mama(X, Y), write(X), write(' is the second mama of '), write(Y), nl.

find\_relationship(X, Y) :-

sec\_sec\_mama(X, Y), write(X), write(' is the seocnd second mama of '), write(Y), nl.

find\_relationship(X, Y) :-

aunt(X, Y), write(X), write(' is the aunt of '), write(Y), nl.

find\_relationship(X, Y) :-

mat\_aunt(X, Y), write(X), write(' is the maternal aunt of '), write(Y), nl.

find\_relationship(X, Y) :-

sec\_mat\_aunt(X, Y), write(X), write(' is the maternal aunt of '), write(Y), nl.

find\_relationship(X, Y) :-

sec\_sec\_mat\_aunt(X, Y), write(X), write(' is th maternal aunt of '), write(Y), nl.

find\_relationship(X, Y) :-

aunt(X, Y), write(X), write(' is the aunt of '), write(Y), nl.

find\_relationship(X, Y) :-

cousin\_sibling(X, Y), write(X), write(' is the cousin of '), write(Y), nl.

find\_relationship(X, Y) :-

sec\_cousin\_sibling(X, Y), write(X), write(' is the cousin of '), write(Y), nl.

find\_relationship(X, Y) :-

cousin(X, Y), write(X), write(' is the cousin of '), write(Y), nl.

find\_relationship(X, Y) :-

deerani(X, Y), write(X), write(' is the deerani of '), write(Y), nl.

find\_relationship(X, Y) :-

deer(X, Y), write(X), write(' is the deer of '), write(Y), nl.

find\_relationship(X, Y) :-

second\_cousin(X, Y), write(X), write(' is the second cousin of '), write(Y), nl.

find\_relationship(X, Y) :-

father\_in\_law(X, Y), write(X), write(' is the father-in-law of '), write(Y), nl.

find\_relationship(X, Y) :-

wife\_of\_brother\_of\_husband(X, Y), write(X), write(' is the wife\_of\_brother\_of\_husband of '), write(Y), nl.

find\_relationship(X, Y) :-

sec\_sec\_grand\_mother(X, Y), write(X), write(' is the second second grand mother of '), write(Y), nl.

find\_relationship(X, Y) :-

sec\_father\_in\_law(X, Y), write(X), write(' is the second father-in-law of '), write(Y), nl.

find\_relationship(X, Y) :-

mother\_in\_law(X, Y), write(X), write(' is the mother-in-law of '), write(Y), nl.

find\_relationship(X, Y) :-

great\_grandfather(X, Y), write(X), write(' is the great-grandfather of '), write(Y), nl.

find\_relationship(X, Y) :-

great\_grandmother(X, Y), write(X), write(' is the great-grandmother of '), write(Y), nl.

find\_relationship(X, Y) :-

grandchild(X, Y), write(X), write(' is the grabdchild of '), write(Y), nl.

find\_relationship(X, Y) :-

husband(X, Y), write(X), write(' is the husband of '), write(Y), nl.

find\_relationship(X, Y) :-

great\_grandchild(X, Y), write(X), write(' is the great grand child of '), write(Y), nl.

find\_relationship(X, Y) :-

sec\_great\_grandfather(X, Y), write(X), write(' is the second great grand father of '), write(Y), nl.

find\_relationship(X, Y) :-

sec\_sister\_in\_law(X, Y), write(X), write(' is the second sister in law of '), write(Y), nl.

find\_relationship(X, Y) :- write('Relationship not defined between '), write(X), write(' and '), write(Y), nl.

Output

