NAME: Soham Phadke  
CLASS: TY\_CS\_D

BATCH: 2

ROLL NO: 37

PRN NO: 12111222

**ASSIGNMENT 5**

**Problem Statement:** Family Tree: Prolog

**Code:**

% Facts: Define relationships in the family tree

male(laxman).

male(parshuram).

male(sumant).

male(dilip).

male(shyam).

male(chintamani).

male(nikhil).

male(saurabh).

male(yogesh).

male(tushar).

male(sharmad).

male(advait).

male(soham).

male(aryan).

male(aditya).

female(suniti).

female(manasi).

female(rekha).

female(leeni).

female(sneha).

female(aditi).

female(anushree).

female(sonali).

female(gauri).

female(girija).

female(aparna).

female(roohi).

female(anvi).

female(asmi).

parent(laxman, parshuram).

parent(suniti, parshuram).

parent(laxman, rekha).

parent(suniti, rekha).

parent(laxman, leeni).

parent(suniti, leeni).

parent(laxman, shyam).

parent(suniti, shyam).

parent(parshuram, chintamani).

parent(manasi, chintamani).

parent(parshuram, nikhil).

parent(manasi, nikhil).

parent(sumant, saurabh).

parent(rekha, saurabh).

parent(sumant, sonali).

parent(rekha, sonali).

parent(dilip,aparna).

parent(leeni,aparna).

parent(shyam,gauri).

parent(sneha,gauri).

parent(shyam,girija).

parent(sneha,girija).

parent(chintamani,advait).

parent(aditi,advait).

parent(chintamani,soham).

parent(aditi,soham).

parent(nikhil,aryan).

parent(anushree,aryan).

parent(nikhil,roohi).

parent(anushree,roohi).

parent(sonali,anvi).

parent(yogesh,anvi).

parent(aparna,asmi).

parent(tushar, asmi).

parent(gauri,aditya).

married(laxman, suniti).

married(parshuram, manasi).

married(sumant, rekha).

married(dilip, leeni).

married(shyam, sneha).

married(chintamani, aditi).

married(nikhil, anushree).

married(yogesh, sonali).

married(tushar, aparna).

married(girija, sharmad).

married(suniti, laxman).

married(manasi, parshuram).

married(rekha, sumant).

married(leeni, dilip).

married(sneha, shyam).

married(aditi, chintamani).

married(anushree, nikhil).

married(sonali, yogesh).

married(aparna, tushar).

married(sharmad, girija).

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

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

wife(X, Y) :- female(X), married(Y, X).

husband(X,Y) :- male(X), married(X,Y).

father\_in\_law(X, Y) :- father(X, Z), married(Z, Y).

mother\_in\_law(X, Y) :- mother(X, Z), married(Z, Y).

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).

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

grandparent(X, Y) :- parent(Z, Y), parent(X, 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), parent(X, Z), female(X).

second\_uncle(X, Y) :- male(X), parent(W, Y), cousin(X, W).

second\_uncle(X, Y) :- male(X), parent(W, Y), cousin(Z, W), married(X, Z).

second\_aunt(X, Y) :- female(X), parent(W, Y), cousin(X, W).

second\_aunt(X, Y) :- female(X), parent(W, Y), cousin(Z, W), married(Z, X).

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

nephew(X, Y) :- male(X), parent(Z, X), sibling(Z, V), married(Y, V).

niece(X, Y) :- female(X), parent(Z, X), sibling(Z, Y).

niece(X, Y) :- female(X), parent(Z, X), sibling(Z, V), married(Y, V).

second\_nephew(X, Y) :- male(X), parent(W, X), cousin(Y, W).

second\_nephew(X, Y) :- male(X), parent(W, X), cousin(V, W), married(Y, V).

second\_niece(X, Y) :- female(X), parent(W, X), cousin(Y, W).

second\_niece(X, Y) :- female(X), parent(W, X), cousin(V, W), married(Y, V).

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

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

second\_grandfather(X, Y) :- grandparent(W, Y), sibling(W, V), male(X), married(X, V).

second\_grandmother(X, Y) :- grandparent(W,Y), sibling(W, X), female(X).

second\_grandmother(X, Y) :- grandparent(W,Y), sibling(W, V), female(X), married(X, V).

great\_grandfather(X, Z) :- grandparent(Y, Z), parent(X, Y), male(X).

great\_grandmother(X, Z) :- grandparent(Y, Z), parent(X, Y), female(X).

sister\_in\_law(X, Y) :- female(X), sibling(Z, Y), married(X, Z).

brother\_in\_law(X, Y) :- male(X), sibling(Z, Y), married(X, Z).

co\_sister\_in\_law(X, Y) :- female(X), married(X, Z), sibling(Z, W), married(W, Y).

% 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) :-

    husband(X, Y), write(X), write(' is the husband 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) :-

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

find\_relationship(X, Y) :-

    grandmotherqq2a1wx(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) :-

    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) :-

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

find\_relationship(X, Y) :-

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

find\_relationship(X, Y) :-

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

find\_relationship(X, Y) :-

    second\_nephew(X, Y), write(X), write(' is the second nephew of '), write(Y), nl.

find\_relationship(X, Y) :-

    second\_niece(X, Y), write(X), write(' is the second niece 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) :-

    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) :-

    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) :-

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

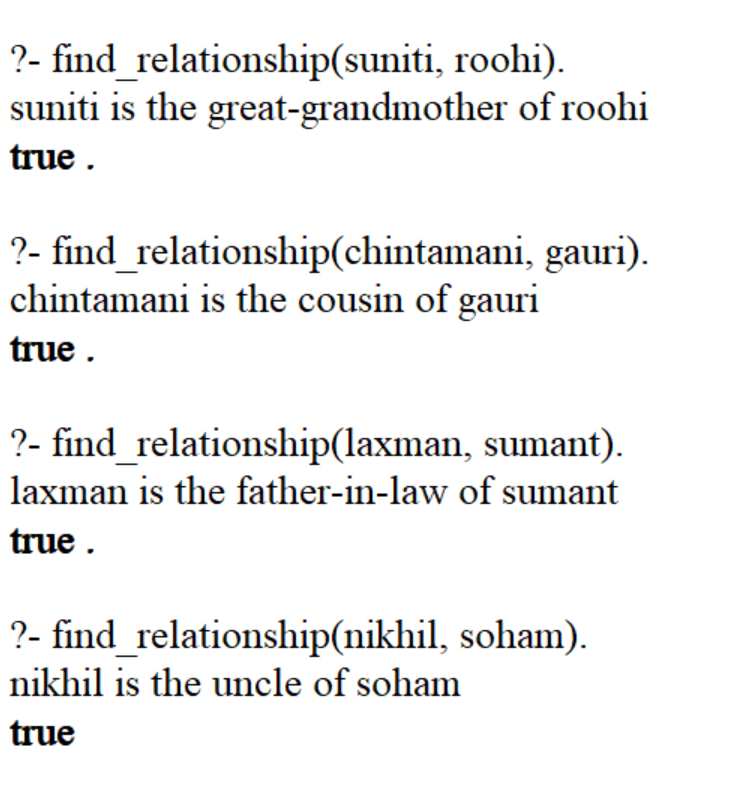
find\_relationship(X, Y) :-

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

find\_relationship(X, Y) :-

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

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

**Output:  
**