**Name**: Suyash Shinde

**Class**: TY -CS   **Div**: D

**Roll no**:28

**Assignment 5:** Implementation predicate logic using PROLOG

**Code**:

male(rajaram).

male(devidas).

male(shivaji).

male(gorakh).

male(gokul).

male(shashi).

male(aba).

male(keda).

male(ramu).

male(ashish).

male(suyash).

male(raj).

male(sahil).

male(krishna).

male(aditya).

male(piyush).

female(tulsabai).

female(ganga).

female(parvati).

female(bharti).

female(hemlata).

female(asmita).

female(nita).

female(ratna).

female(chaya).

female(punam).

female(teju).

female(rani).

female(gauri).

female(sidhi).

female(mansi).

parent(rajaram,devidas).

parent(rajaram,shivaji).

parent(tulsabai,devidas).

parent(tulsabai,shivaji).

parent(devidas,gorakh).

parent(devidas,gokul).

parent(devidas,shashi).

parent(devidas,aba).

parent(ganga,gorakh).

parent(ganga,gokul).

parent(ganga,shashi).

parent(ganga,aba).

parent(shivaji,keda).

parent(shivaji,chaya).

parent(shivaji,punam).

parent(parvati,keda).

parent(parvati,chaya).

parent(parvati,punam).

parent(gorakh,suyash).

parent(bharti,suyash).

parent(gokul,teju).

parent(gokul,raj).

parent(gokul,sahil).

parent(hemlata,teju).

parent(hemlata,raj).

parent(hemlata,sahil).

parent(shashi,rani).

parent(asmita,rani).

parent(aba,gauri).

parent(aba,krishna).

parent(nita,gauri).

parent(nita,krishna).

parent(keda,aditya).

parent(ratna,aditya).

parent(ramu,sidhi).

parent(ramu,mansi).

parent(chaya,sidhi).

parent(chaya,mansi).

parent(ashish,piyush)

parent(punam,piyush).

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

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

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

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

secondfather\_in\_law(X, Y) :- male(X), father\_in\_law(Z, Y), sibling(X, Z).

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

secondmother\_in\_law(X, Y) :- female(X),female(Y),mother\_in\_law(Z, Y), sister\_in\_law(Z, X).

son\_in\_law(X, Y) :- male(X), wife(Z, X), father(Y, Z).

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).

brother\_in\_law(X,Y) :- male(W),male(X),sibling(X,W),father(W,T),mother(Y,T).

brother\_in\_law2(X,Y) :- male(X), male(Y), sibling(Z, X), wife(Z, Y).

sister\_in\_law(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).

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

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

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

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

% 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) :-

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) :-

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) :-

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

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

find\_relationship(X, Y) :-

brother\_in\_law2(X, Y), write(X), write(' is the brother\_in\_law of '), write(Y), nl.

find\_relationship(X, Y) :-

son\_in\_law(X, Y), write(X), write(' is the son\_in\_law 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) :-

secondfather\_in\_law(X, Y), write(X), write(' is the secondfather-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) :-

secondmother\_in\_law(X, Y), write(X), write(' is the secondmother-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) :-

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

find\_relationship(X, Y) :-

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