**Name**: Sakshi Shewale

**Class**: TY -CS   **Div**: D

**Roll no**:24

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

**Code**:

% Facts: Define relationships in the family tree

male(namdeo).

male(jairam).

male(motiram).

male(sanjay).

male(mahesh).

male(umesh).

male(dinesh).

male(milind).

male(pratik).

male(shivam).

male(prasad).

male(gaurav).

female(rukhmini).

female(bandini).

female(lata).

female(swati).

female(yogita).

female(kavita).

female(preeti).

female(aparna).

female(sakshi).

female(ishwari).

female(aaradhya).

female(kuhu).

parent(namdeo, jairam).

parent(namdeo, motiram).

parent(jairam, sanjay).

parent(jairam, mahesh).

parent(jairam, mahesh).

parent(motiram, preeti).

parent(motiram, dinesh).

parent(rukhmini, jairam).

parent(rukhmini,motiram).

parent(bandini,sanjay).

parent(bandini,mahesh).

parent(bandini,umesh).

parent(lata,preeti).

parent(lata,dinesh).

parent(sanjay,pratik).

parent(mahesh,sakshi).

parent(mahesh,shivam).

parent(umesh,prasad).

parent(swati,pratik).

parent(yogita,sakshi).

parent(yogita,shivam).

parent(kavita,prasad).

parent(milind,ishwari).

parent(milind,gaurav).

parent(preeti,ishwari).

parent(preeti,gaurav).

parent(dinesh,aaradhya).

parent(dinesh,kuhu).

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).

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).

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

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),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).

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).

% 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) :-

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

**Output**:

