**PRACTICAL-9**

**AIM:** Assume given a set of facts of the form father(name1,name2) (name1 is the father of name2)

Define a predicate cousin(X,Y) which holds iff X and Y are cousins.

Define a predicate grandson(X,Y) which holds iff X is a grandson of Y.

Define a predicate descendent(X,Y) which holds iff X is a descendent of Y. Define a predicate grandparent(X,Y) which holds iff X is a grandparent of Y. Consider the following genealogical tree:

father(a,b). father(a,c). father(b,d). father(b,e). father(c,f).

Say which answers, and in which order, are generated by your definitions for the following queries in Prolog:

?- cousin(X,Y).

?- grandson(X,Y).

?- descendent(X,Y). ?-grandparent(X,Y).

**Source Code:**

father(a, b).

father(a, c).

father(b, d).

father(b, e).

father(c, f).

cousin(X, Y) :-

father(P1, X),

father(P2, Y),

P1 \= P2, % Ensure they have different fathers

siblings(P1, P2). % Check if their fathers are siblings

siblings(X, Y) :-

father(F, X),

father(F, Y),

X \= Y.

grandson(X, Y) :-

father(Y, Z),

father(Z, X).

descendant(X, Y) :-

father(Y, X).

descendant(X, Y) :-

father(Z, X),

descendant(Z, Y).

grandparent(X, Y) :-

father(X, Z),

father(Z, Y).

**Output:**

