LAB:2 INTRODUCTION TO PROLOG

1. **Making .pl file and consulting.**

* At first make a simple text file and rename it to test.pl
* Write in file

boy(ramesh)

Through the Prolog Terminal, use the command

?- consult (<path of the file with .pl file>).

?-consult (D:/LAB/AI/test.pl)

Then Try:

?- boy(ramesh).

TRUE

The file can be directly consulted form File>Consult>Select File.

Filename : first.pl

boy(ramesh).

girl(rama).

Terminal:

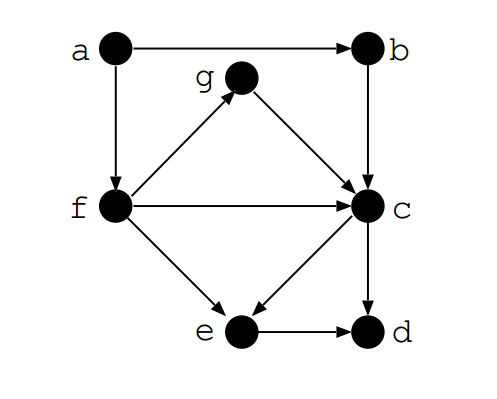
1 ?- boy(ramesh). // ramesh is a boy. So the program returns true after consulting

true.

2 ?- boy(rama). //rama is not a boy. So the program returns false after consulting

false.

1. **Graph**



**Implement the given graph.**

File name: graph.pl

edge(a,b). // a vertex is connected to vertex b

edge(a,f). // a vertex is connected to vertex f

edge(f,e). // f vertex is connected to vertex e

edge(e,d). // e vertex is connected to vertex d

edge(b,c). // c vertex is connected to vertex c

edge(f,c). // f vertex is connected to vertex c

edge(c,e). // c vertex is connected to vertex e

edge(f,g). // f vertex is connected to vertex g

edge(g,c). // g vertex is connected to vertex c

edge(c,d) // c vertex is connected to vertex d

path(Node1,Node2):- edge(Node1,Node2). /\*If node1 and node 2 is connected, then there is path between them.\*/

path(Node1, Node2) :- edge(Node1,Somenode),path(Somenode,Node2). /\* This is the recursion that checks the path between two nodes which are connected through many edges \*/

Terminal:

1 ?- edge(a,b).

true .

2 ?- edge(X,c).

X = b ;

X = f ;

X = g.

3 ?- path(a,f).

true .

4 ?- path(a,d).

true ;

1. **Family Representation**

File name: family.pl

male(amar).

male(chandra).

female(bina).

female(divya).

parent(amar,chandra).

parent(amar,divya).

parent(bina,chandra).

parent(bina,divya).

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

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

/\* X is sibling of Y if parents are same and X and Y are different \*/

sibling(X,Y):- parent(Z,X),parent(Z,Y),different(X,Y).

/\* X is not different to X \*/

different(X,X):- !,fail.

/\* X is different than Y \*/

different(X,Y).

Terminal

?- consult('family.pl').  
true.

?- mother(bina, divya).  
true.

?- father(amar, divya).  
true.

?- father(amar, chandra).  
true .

?- sibling(chandra, chandra).  
false.

?- sibling(chandra, divya).  
true .

?- parent(X, chandra).  
X = amar ;  
X = bina.

?- sibling(amar, chandra).  
false.