**Practical No : 02**

**Problem Statement:**

1) Write a Prolog Program for parent relationship.

2) Write a Prolog program to assert parent relationship along with gender facts. ( Hint: Use predicates: parent, male, female OR parent, gender) - Use family tree provided at slide no-17 to assert facts.

3) Write a Prolog program for directed graph-1. (Use graph from slide-19.)

4) Write a Prolog program for directed graph-2. (as per slide-20 (A))

5) Write a Prolog program for undirected graph. ( as per slide-20 (B))

1) Write a Prolog Program for parent relationship.

**Code:-**

parent(paul,petunia).

parent(paul,lili).

parent(helen,petunia).

parent(helen,lili).

parent(albert,james).

parent(ruth,james).

parent(vernom,dudley).

parent(petunia,dudley).

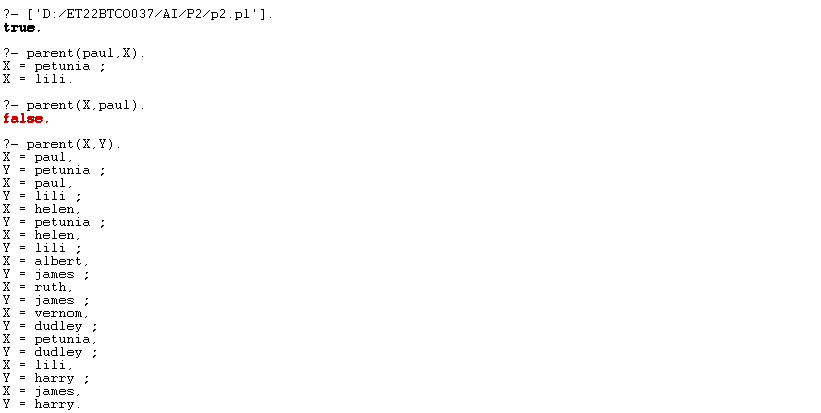
parent(lili,harry).

parent(james,harry).

**Output:-**







2) Write a Prolog program to assert parent relationship along with gender facts. ( Hint: Use predicates: parent, male, female OR parent, gender) - Use family tree provided at slide no-17 to assert facts.

**Code:-**

parent(paul,petunia).

parent(paul,lili).

parent(helen,petunia).

parent(helen,lili).

parent(albert,james).

parent(ruth,james).

parent(vernom,dudley).

parent(petunia,dudley).

parent(lili,harry).

parent(james,harry).

male(paul).

male(albert).

male(james).

male(vernon).

male(harry).

female(helen).

female(ruth).

female(lili).

female(petunia).

**Output:-**





3) Write a Prolog program for directed graph-1. (Use graph from slide-19.)

**Code:-**

edge(a,b).

edge(b,c).

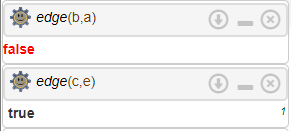
edge(c,e) .

edge(e,f) .

edge(e,d).

edge(d,b).

**Output:-**





4) Write a Prolog program for directed graph-2. (as per slide-20 (A))

**Code:-**

edge(c,e).

edge(d,e).

edge(e,a).

edge(c,d).

edge(a,d).

edge(a,c).

edge(a,b).

edge(b,d).

**Output:-**





5) Write a Prolog program for undirected graph. ( as per slide-20 (B))

**Code:-**

edge(a,b).

edge(b,a).

edge(c,b).

edge(b,c).

edge(c,e).

edge(e,c).

edge(c,d).

edge(d,c).

edge(e,d).

edge(d,e).

edge(e,f).

edge(f,e).

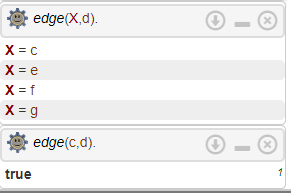
edge(f,d).

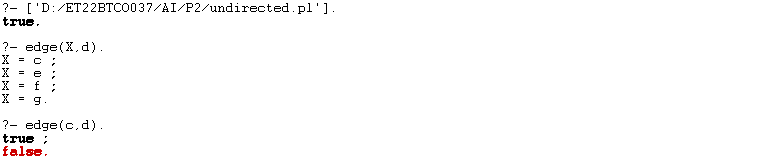
edge(d,f).

edge(d,g).

edge(g,d).

**Output:-**





****

