

EX.NO: 07**DATE:****REG.NO: 220701065****PROLOG- FAMILY TREE****AIM**

To develop a family tree program using PROLOG with all possible facts, rules, and queries.

SOURCE CODE:**KNOWLEDGE BASE:**

```
/*FACTS :: */
```

```
male(peter).
```

```
male(john). male(chris).
```

```
male(kevin).
```

```
female(betty).
```

```
female(jeny). female(lisa).
```

```
female(helen).
```

```
parentOf(chris,peter).
```

```
parentOf(chris,betty).
```

```
parentOf(helen,peter).
```

```
parentOf(helen,betty).
```

```
parentOf(kevin,chris).
```

```
parentOf(kevin,lisa). parentOf(jeny,john).
```

```
parentOf(jeny,helen).
```

```
/*RULES :: */
```

```
/* son,parent
```

```
* son,grandparent*/
```

```
father(X,Y):- male(Y), parentOf(X,Y).
```

```
mother(X,Y):- female(Y), parentOf(X,Y).
```

```
grandfather(X,Y):- male(Y),parentOf(X,Z),parentOf(Z,Y).
```

```
grandmother(X,Y):- female(Y),parentOf(X,Z),parentOf(Z,Y).
```

```
brother(X,Y):- male(Y), father(X,Z), father(Y,W),Z==W.
```

sister(X,Y):- female(Y), father(X,Z),father(Y,W),Z==W.

OUTPUT:

The screenshot shows a Prolog interpreter window with the following queries and results:

- Query: `male(peter).` Result: `true`
- Query: `father(chris,peter).` Result: `true`
- Query: `father(chris,betty)` Result: `false`
- Query: `mother(chris,X).` Result: `X = betty`
- Query: `brother(chris,helen).` Result: `false`

At the bottom, there is a query prompt `?- brother(chris,helen).` and a status bar with buttons for `Examples`, `History`, `Solutions`, a checkbox for `table results`, and a `Run!` button.

RESULT:

Thus the family tree program using PROLOG with all possible facts, rules, and queries has been implemented successfully.