EX.NO :

DATE :

PROLOG

AIM :

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

SOURCE CODE:

```prolog
1  male(peter).
2  male(john).
3  male(chris).
4  male(kevin).
5
6  female(betty).
7  female(jeny).
8  female(lisa).
9  female(helen).
10
11 parentOf(chris,peter).
12 parentOf(chris,betty).
13 parentOf(helen,peter).
14 parentOf(helen,betty).
15 parentOf(kevin,chris).
16 parentOf(kevin,lisa).
17 parentOf(jeny,john).
18 parentOf(jeny,helen).
19
20 father(X,Y):- male(Y), parentOf(X,Y).
21 mother(X,Y):- female(Y), parentOf(X,Y).
22 grandfather(X,Y):- male(Y),parentOf(X,Z),parentOf(Z,Y).
23 grandmother(X,Y):- female(Y),parentOf(X,Z),parentOf(Z,Y).
24 brother(X,Y):- male(Y), father(X,Z), father(Y,W),Z==W.
25 sister(X,Y):- female(Y), father(X,Z),father(Y,W),Z==W.
```

OUTPUT :

```
?- sister(X,Y).
```

Examples▴ History▴ Solutions▴   ☐ table results Run!

father(X,Y).   ⊕ — ⊗
X = chris,
Y = peter
Next 10 100 1,000 Stop

mother(X,Y).   ⊕ — ⊗
X = chris,
Y = betty
Next 10 100 1,000 Stop

sister(X,Y).   ⊕ — ⊗
X = Y, **Y** = jeny
Next 10 100 1,000 Stop

RESULT :

Thus the program is successfully executed and output is verified