

EX.NO : 6**PROLOG****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 :

male(peter)	⊞	⊞
true		⊞
father(chris,peter)	⊞	⊞
true		⊞
father(chris,betty)	⊞	⊞
false		⊞
grandfather(kevin,peter)	⊞	⊞
true		⊞
grandfather(jerry,peter)	⊞	⊞
true		⊞
grandmother(jerry,peter)	⊞	⊞
false		⊞
mother(chris,X)	⊞	⊞
X = betty		⊞
brother(helen,chris)	⊞	⊞
true		⊞
brother(chris,helen)	⊞	⊞
false		⊞
father(X,Y)	⊞	⊞
X = chris, Y = peter X = helen, Y = peter X = jerry, Y = john X = kevin, Y = chris		⊞
mother(X,Y)	⊞	⊞
X = chris, Y = betty X = helen, Y = betty X = kevin, Y = lisa X = jerry, Y = helen		⊞

```
grandmother(X,Y)
X = kevin,
Y = betty
X = jeny,
Y = betty

grandfather(X,Y)
X = kevin,
Y = peter
X = jeny,
Y = peter
```

```
brother(X,Y)
X = Y, Y = chris
X = helen,
Y = chris
X = Y, Y = kevin

sister(X,Y)
X = Y, Y = jeny
X = chris,
Y = helen
X = Y, Y = helen
```

RESULT :

Thus the python code is implemented successfully and the output is verified.