

EX.NO: 07

DATE

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:

 <code>male(peter)</code>	 
<code>true</code>	
 <code>father(chris,peter)</code>	 
<code>true</code>	
 <code>father(chris,betty)</code>	 
<code>false</code>	
 <code>grandfather(kevin,peter)</code>	 
<code>true</code>	
 <code>grandfather(jeny,peter)</code>	 
<code>true</code>	
 <code>grandmother(jeny,peter)</code>	 
<code>false</code>	
 <code>mother(chris,X)</code>	 
<code>X = betty</code>	
 <code>brother(helen,chris)</code>	 
<code>true</code>	
 <code>brother(chris,helen)</code>	 
<code>false</code>	
 <code>father(X,Y)</code>	 
<code>X = chris,</code>	
<code>Y = peter</code>	
<code>X = helen,</code>	
<code>Y = peter</code>	
<code>X = jeny,</code>	
<code>Y = john</code>	
<code>X = kevin,</code>	
<code>Y = chris</code>	
 <code>mother(X,Y)</code>	 
<code>X = chris,</code>	
<code>Y = betty</code>	
<code>X = helen,</code>	
<code>Y = betty</code>	
<code>X = kevin,</code>	
<code>Y = lisa</code>	
<code>X = jeny,</code>	
<code>Y = helen</code>	
 <code>grandmother(X,Y)</code>	 
<code>X = kevin,</code>	
<code>Y = betty</code>	
<code>X = jeny,</code>	
<code>Y = betty</code>	
 <code>grandfather(X,Y)</code>	 
<code>X = kevin,</code>	
<code>Y = peter</code>	
<code>X = jeny,</code>	
<code>Y = peter</code>	
 <code>brother(X,Y)</code>	 
<code>X = Y, Y = chris</code>	
<code>X = helen,</code>	
<code>Y = chris</code>	
<code>X = Y, Y = kevin</code>	
 <code>sister(X,Y)</code>	 
<code>X = Y, Y = jeny</code>	
<code>X = chris,</code>	
<code>Y = helen</code>	
<code>X = Y, Y = helen</code>	

RESULT:

Thus, developing a family tree program using PROLOG with all possible facts, rules, and queries is successful.