

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:



365 users online


male(Y), parentOf(X,Y).


⬇

⌵

✕

Y	X	
peter	chris	
peter	helen	2
john	jeny	3
chris	kevin	4

false



female(Y), parentOf(X,Y).

⬇

⌵

✕

Y	X	
betty	chris	1
betty	helen	2
lisa	kevin	3
helen	jeny	4


male(Y),parentOf(X,Z),parentOf(Z,Y).


⬇

⌵

✕

Y	X	Z	
peter	kevin	chris	1
peter	jeny	helen	2

false


female(Y),parentOf(X,Z),parentOf(Z,Y).


⬇

⌵

✕

Y	X	Z	
betty	kevin	chris	1
betty	jeny	helen	2

false



male(Y), father(X,Z), father(Y,W),Z==W.

⬇

⌵

✕

procedure `father(A,B)` does not exist


female(Y), father(X,Z),father(Y,W),Z==W.

⬇

⌵

✕

procedure `father(A,B)` does not exist

?

Examples

History

Solutions

☒ table results

Run

RESULT:

Thus, we have developed a family tree program using PROLOG with all possible facts, rules, and queries.