

EX.NO: 07

DATE:

REG.NO: 220701517

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).
```

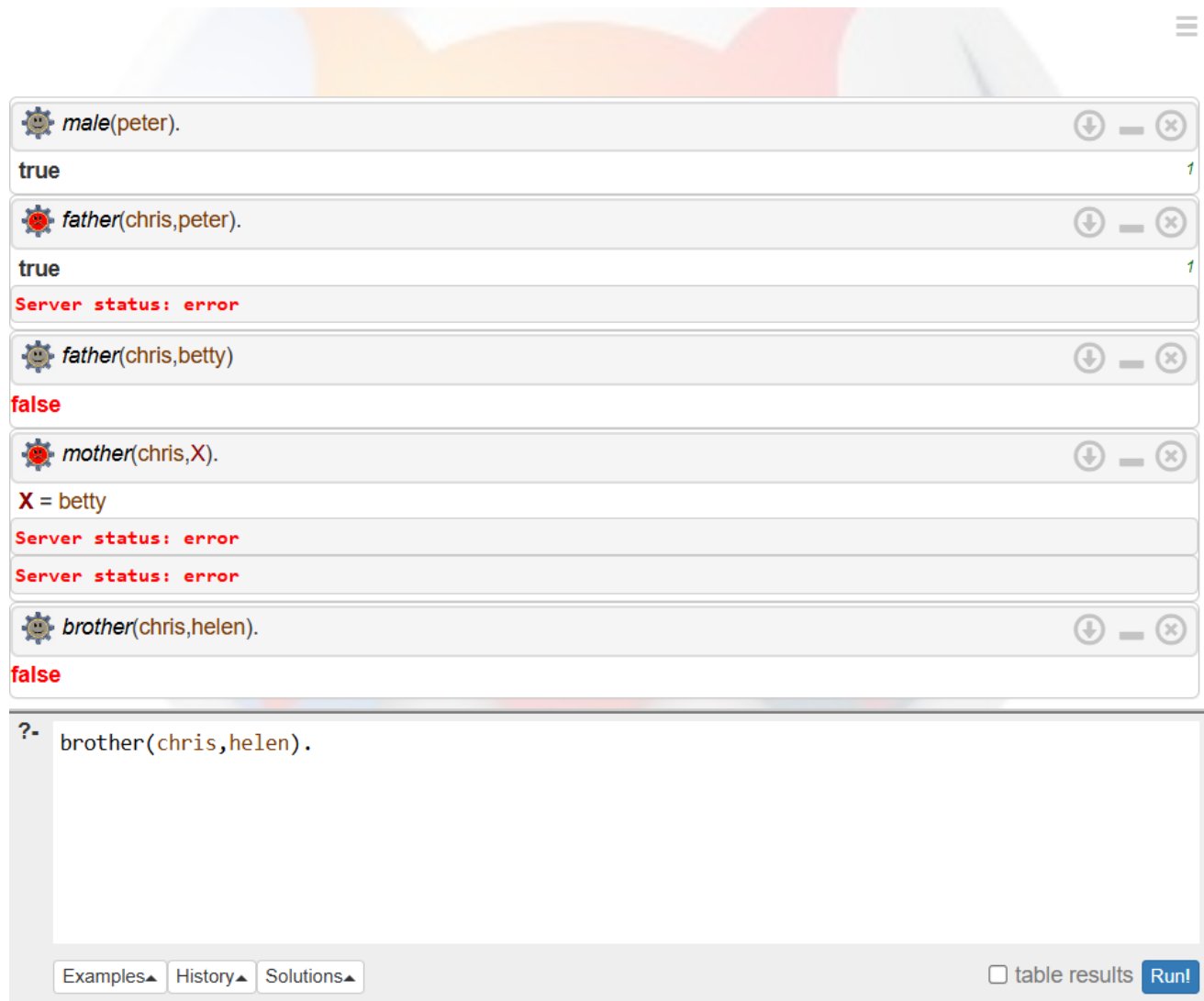
```
grandfather(X,Y):- male(Y),parentOf(X,Z),parentOf(Z,Y).
```

```
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 a list of queries and their results. Each query is in a separate row with a gear icon, a status icon (green for success, red for error), and window controls (download, close, maximize). The results are displayed below each query.

Query	Result
<code>male(peter).</code>	true
<code>father(chris,peter).</code>	true
<code>father(chris,betty)</code>	false
<code>mother(chris,X).</code>	X = betty Server status: error Server status: error
<code>brother(chris,helen).</code>	false

Below the list of queries, there is a large text area for a new query, currently containing `?- brother(chris,helen).`. At the bottom of the window, there are buttons for "Examples", "History", and "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.