



***Department of Computer Science and Engineering***

***Course Title*** – Artificial Intelligence and Expert Systems || Lab

***Course Code*** – CSE 404

***Assignment*** -Basic family relationship tree structure of family using Prolog.

***Date of Submission-*** 14.02.2023

***Submitted By***

Rifa Tasfia

Reg. No-19201045 Section-A2.

***Submitted To***

Dr. Nasima Begum

Associate Professor, UAP, CSE Department

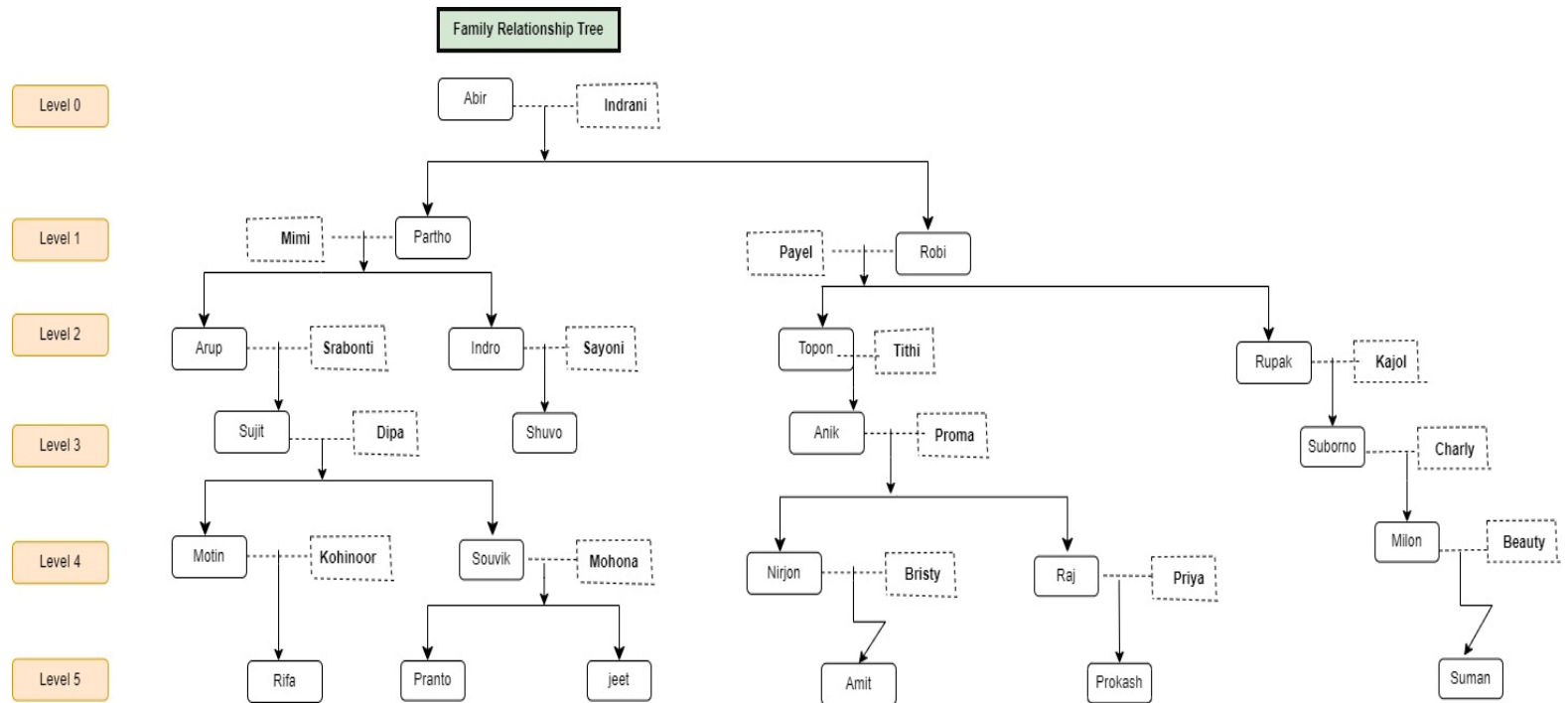
***Problem Title:*** Implement a Basic family Relationship  
Tree Structure of my own family using Prolog.

***Problem Description:*** We need to design the relationship tree structure of my family using “Prolog”. Also write rules against degree and removal for up to 3<sup>rd</sup> degree & twice removed situations for cousin relationships. We have to use recursion in my rules for different family relations.

***Tools & Languages:***

- Diagram.net. (Design Tree).
- Notepad (Write rules & facts).
- SWI Prolog.

## Diagram:



The above diagram is my family relationship tree structure. Now with help of SWI-Prolog I'll remove first, second, third cousin twice times. In the tree, all bold rectangular box represents males and dotted arrow represents wife.

## Necessary Logic:

Relationship between subject and relative given the relationship to their most recent common ancestor

		Relative		
Separation in generations to ancestor R→		2	3	4
S↓	Relationship to ancestor	Grandparent	Great-grandparent	Great-great-grandparent
Subject	2	Grandparent	1st cousin	1st cousin once removed
	3	Great-grandparent	1st cousin once removed	2nd cousin
	4	Great-great-grandparent	1st cousin twice removed	2nd cousin once removed

For cousins ( $R \geq 2$  and  $S \geq 2$ ): degree =  $\min(R, S) - 1$ , removal =  $|R - S|$

**Source Code:** <https://github.com/tasfiarifa/CSE-404-4.1-AI-lab->

**Sample Output:** In the screenshot, here is the sample output for father, mother, parent, grandparent, great-grandparent, great great grand parent for a individual person.

The screenshot shows the SWISH Prolog IDE interface. On the left, a list of facts is defined:

```

/*facts*/
male(abir).
male(partho).
male(robi).
male(arup).
male(indro).
male(rupak).
male(topon).
male(sujit).
male(shuvo).
male(anic).
male(suborno).
male(motin).
male(souvik).
male(nirjon).
male(raj).
male(milon).
male(pranto).
male(jeet).
male(amt).
male(prokash).
male(uman).
female(indrani).
female(mimi).
female(payel).

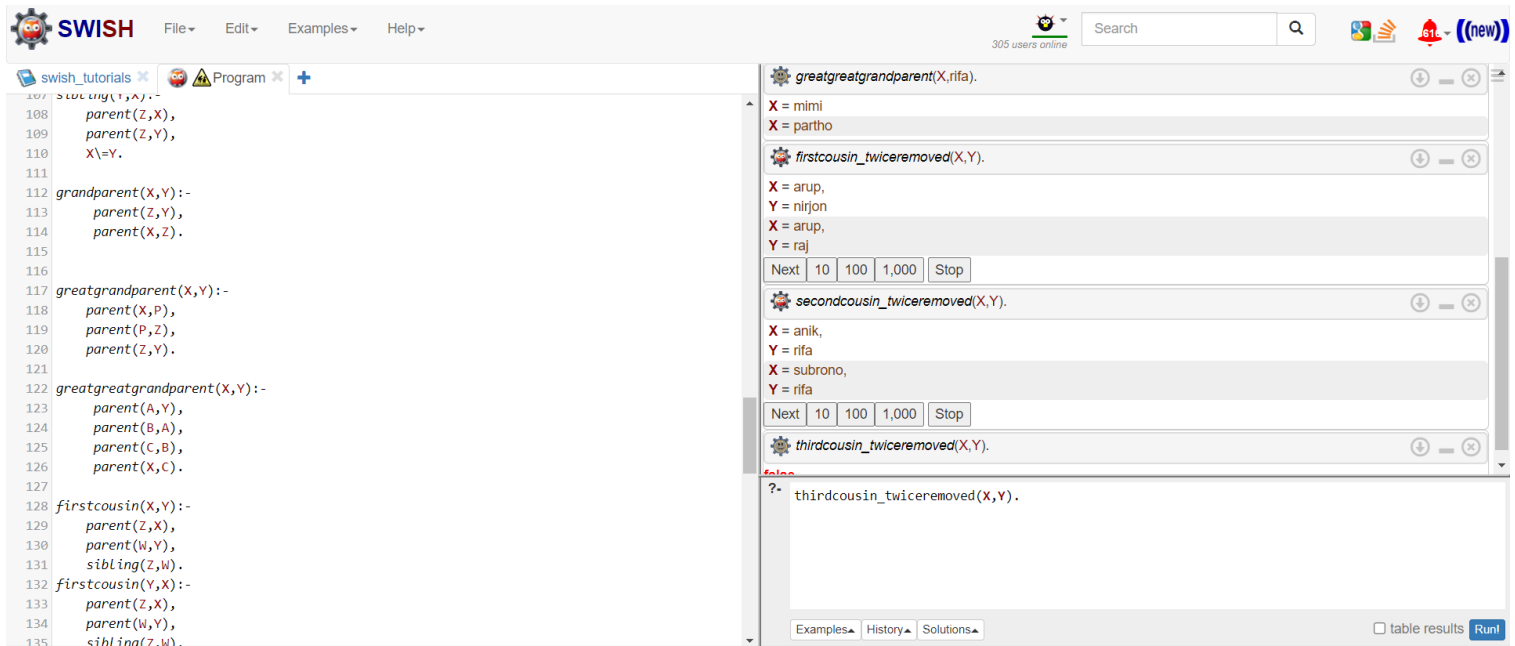
```

On the right, a list of queries is shown, each with its results:

- `father(X,rifa).` results in `X = motin`
- `mother(X,rifa).` results in `X = kohinoor`
- `parent(X,rifa).` results in `X = kohinoor` and `X = motin`
- `grandparent(X,rifa).` results in `X = sujit`
- `greatgrandparent(X,rifa).` results in `X = srabonti` and `X = arup`
- `greatgreatgrandparent(X,rifa).` results in no output.

At the bottom, a query `?- thirdcousin_twiceremoved(X,Y).` is entered, and the interface includes buttons for 'Examples', 'History', 'Solutions', 'table results', and a 'Run!' button.

And here is the sample output by removing first cousin, second cousin and third cousin .



The screenshot shows the SWISH Prolog IDE interface. On the left, a code editor displays a Prolog program with rules for parent, grandparent, greatgrandparent, firstcousin, and twiceremoved relationships. The right pane shows the execution results for the query `greatgreatgrandparent(X,rifa).`, displaying three results for X: mimi, partho, and arup. Below each result, the twiceremoved predicate is shown with its arguments. The interface includes a menu bar, a search bar, and a status bar.

```
107 > load_module('1.pl').
108 parent(Z,X),
109 parent(Z,Y),
110 X\=Y.
111
112 grandparent(X,Y):-
113     parent(Z,Y),
114     parent(X,Z).
115
116
117 greatgrandparent(X,Y):-
118     parent(X,P),
119     parent(P,Z),
120     parent(Z,Y).
121
122 greatgreatgrandparent(X,Y):-
123     parent(A,Y),
124     parent(B,A),
125     parent(C,B),
126     parent(X,C).
127
128 firstcousin(X,Y):-
129     parent(Z,X),
130     parent(W,Y),
131     sibling(Z,W).
132 firstcousin(Y,X):-
133     parent(Z,X),
134     parent(W,Y),
135     sibling(Z,W).
```

Execution results for `greatgreatgrandparent(X,rifa).`:

- X = mimi
- X = partho
- X = arup, Y = nirjon, X = arup, Y = raj
- X = anik, Y = rifa, X = subrono, Y = rifa

Below each result, the twiceremoved predicate is shown with its arguments. The interface includes a menu bar, a search bar, and a status bar.

## Challenges & Conclusion:

Faced some minor difficulties while writing the code. Name and relations should be correctly placed. Facts have to be as simple and specific. There were some errors in SWI-Prolog but those were fixed successfully after some troubleshooting.

To implement any knowledge-base, creating a decision tree and a structured logics can help a lot while troubleshooting. Naming conventions are case sensitive. Be careful of spellings.