

Name: Shawon Barman
Reg ID: 18201043

UNIVERSITY OF ASIA PACIFIC

Department of Computer Science and Engineering



Course Title :

Artificial Intelligence and Expert Systems Lab

Course Code : CSE 404

Assignment No: 01

Submitted By:

Shawon Barman

Section: A

ID: 18201043

Submitted to:

Dr. Nasima Begum

Assistant Professor

Department of CSE, University Of Asia Pacific

Problem Title: Implement a basic family relationship tree structure of your own family using Prolog

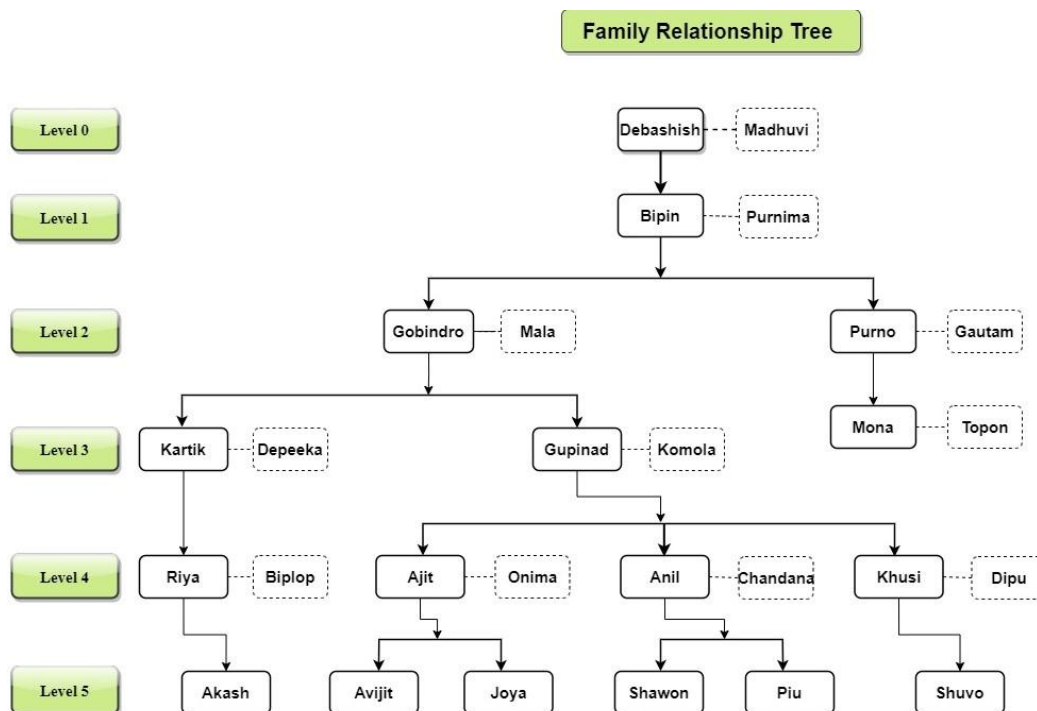
Problem Description:

Implement a basic family tree structure of your own family using prolog. Write the rules against the degree and removal up to the 3rd degree and the twice removed situation.

Tools and Languages:

- Draw.io
- Notepad
- SWI-Prolog

Diagram:



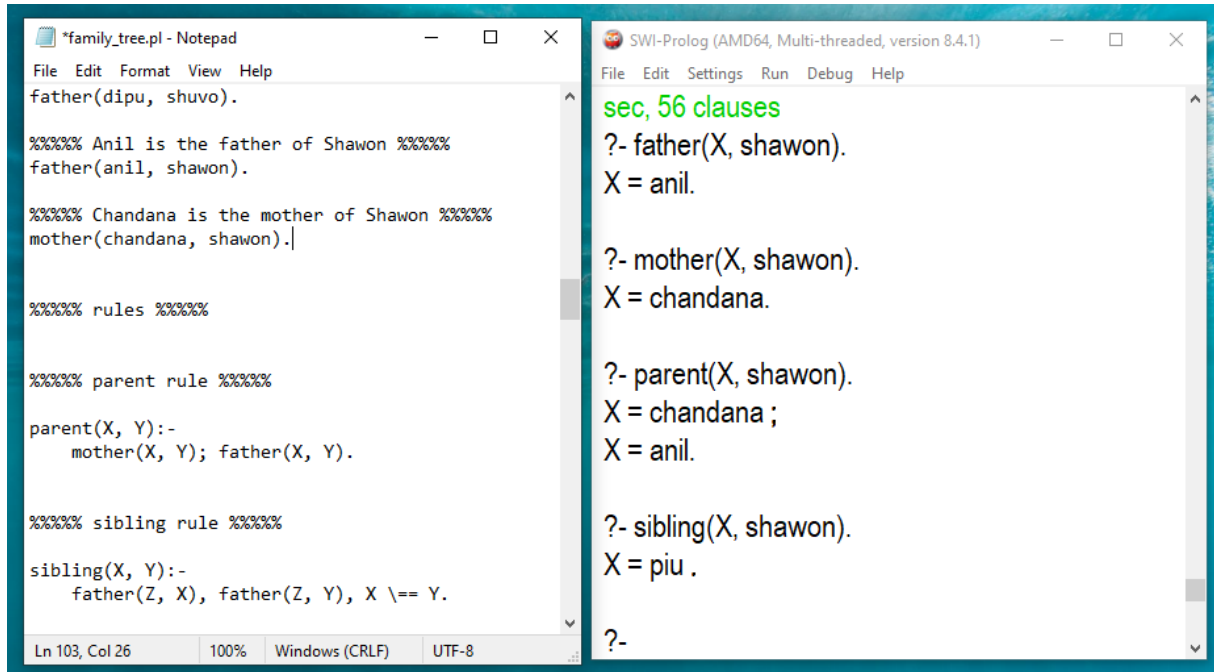
The above diagram is my family relationship tree structure. Now with help of SWI-Prolog, I'll remove the first, second, third cousin twice times. In the tree, all bold rectangular box represents a strong entity, and the dotted rectangular box represents a weak entity.

Name: Shawon Barman

Reg ID: 18201043

Sample Input/Output:

In the below screenshot, here is the sample output for father, mother, parent, and sibling for a individual person.



The screenshot shows two windows. The left window is a Notepad editor titled '*family_tree.pl - Notepad' containing Prolog code. The right window is the SWI-Prolog IDE titled 'SWI-Prolog (AMD64, Multi-threaded, version 8.4.1)' showing the execution of queries.

```
*family_tree.pl - Notepad
File Edit Format View Help
father(dipu, shuvo).

%%%%% Anil is the father of Shawon %%%%%
father(anil, shawon).

%%%%% Chandana is the mother of Shawon %%%%%
mother(chandana, shawon).

%%%%% rules %%%%%

%%%%% parent rule %%%%%
parent(X, Y):-
    mother(X, Y); father(X, Y).

%%%%% sibling rule %%%%%
sibling(X, Y):-
    father(Z, X), father(Z, Y), X \== Y.

Ln 103, Col 26    100%    Windows (CRLF)    UTF-8
```

```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.1)
File Edit Settings Run Debug Help

sec, 56 clauses
?- father(X, shawon).
X = anil.

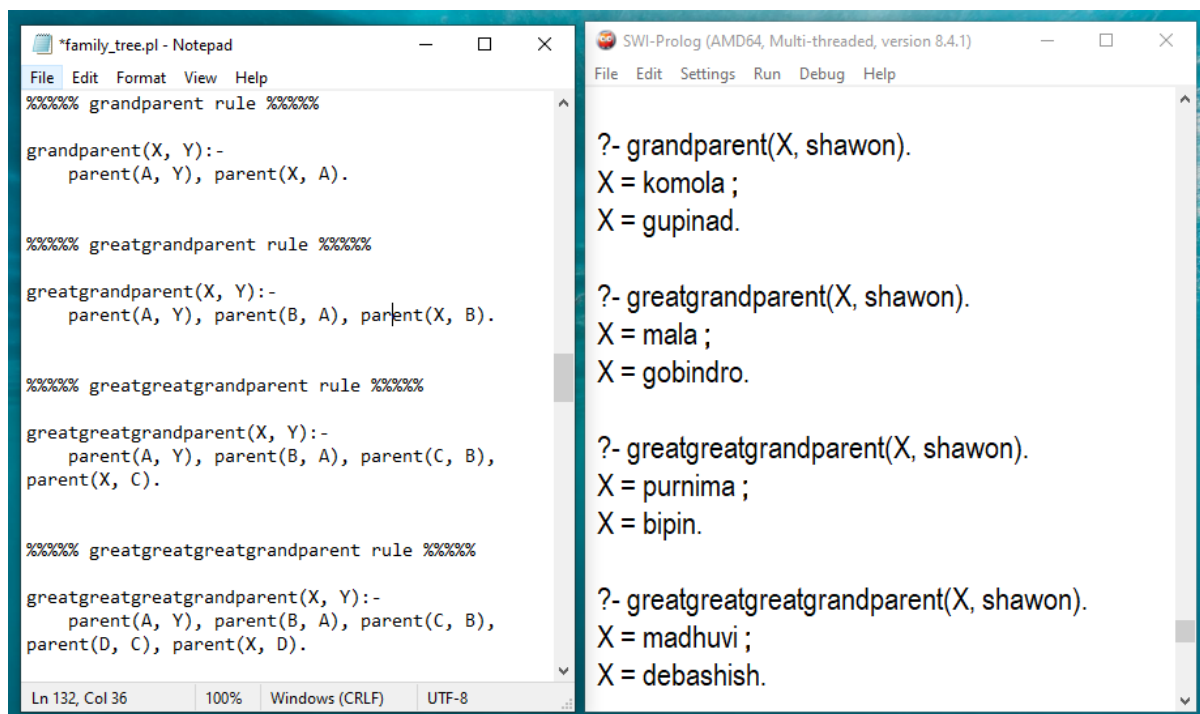
?- mother(X, shawon).
X = chandana.

?- parent(X, shawon).
X = chandana ;
X = anil.

?- sibling(X, shawon).
X = piu .

?-
```

In the below screenshot, here is the sample output for grandparent, great-grandparent, great-great-grandparent, great- great-great-grandparent for a individual person.



The screenshot shows two windows. The left window is a Notepad editor titled '*family_tree.pl - Notepad' containing Prolog code for grandparent and great-grandparent rules. The right window is the SWI-Prolog IDE titled 'SWI-Prolog (AMD64, Multi-threaded, version 8.4.1)' showing the execution of queries.

```
*family_tree.pl - Notepad
File Edit Format View Help
%%%%% grandparent rule %%%%%
grandparent(X, Y):-
    parent(A, Y), parent(X, A).

%%%%% greatgrandparent rule %%%%%
greatgrandparent(X, Y):-
    parent(A, Y), parent(B, A), parent(X, B).

%%%%% greatgreatgrandparent rule %%%%%
greatgreatgrandparent(X, Y):-
    parent(A, Y), parent(B, A), parent(C, B),
    parent(X, C).

%%%%% greatgreatgreatgrandparent rule %%%%%
greatgreatgreatgrandparent(X, Y):-
    parent(A, Y), parent(B, A), parent(C, B),
    parent(D, C), parent(X, D).

Ln 132, Col 36    100%    Windows (CRLF)    UTF-8
```

```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.1)
File Edit Settings Run Debug Help

?- grandparent(X, shawon).
X = komola ;
X = gupinad.

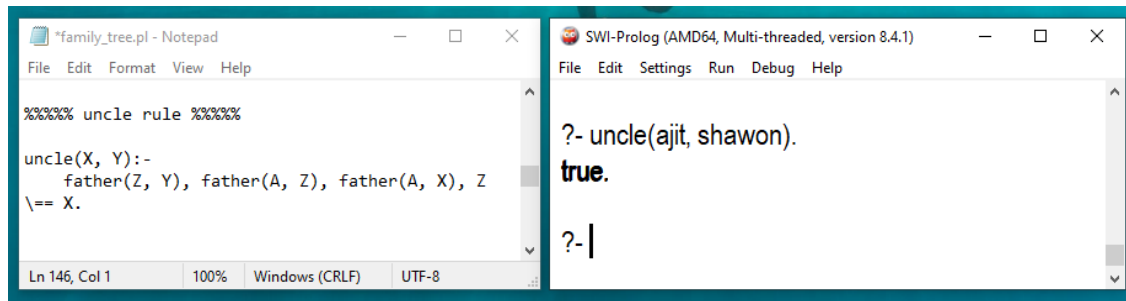
?- greatgrandparent(X, shawon).
X = mala ;
X = gobindro.

?- greatgreatgrandparent(X, shawon).
X = purnima ;
X = bipin.

?- greatgreatgreatgrandparent(X, shawon).
X = madhuvi ;
X = debashish.
```

Name: Shawon Barman
Reg ID: 18201043

Is Ajit is the uncle of shawon? The sample input and output in the below screenshot.



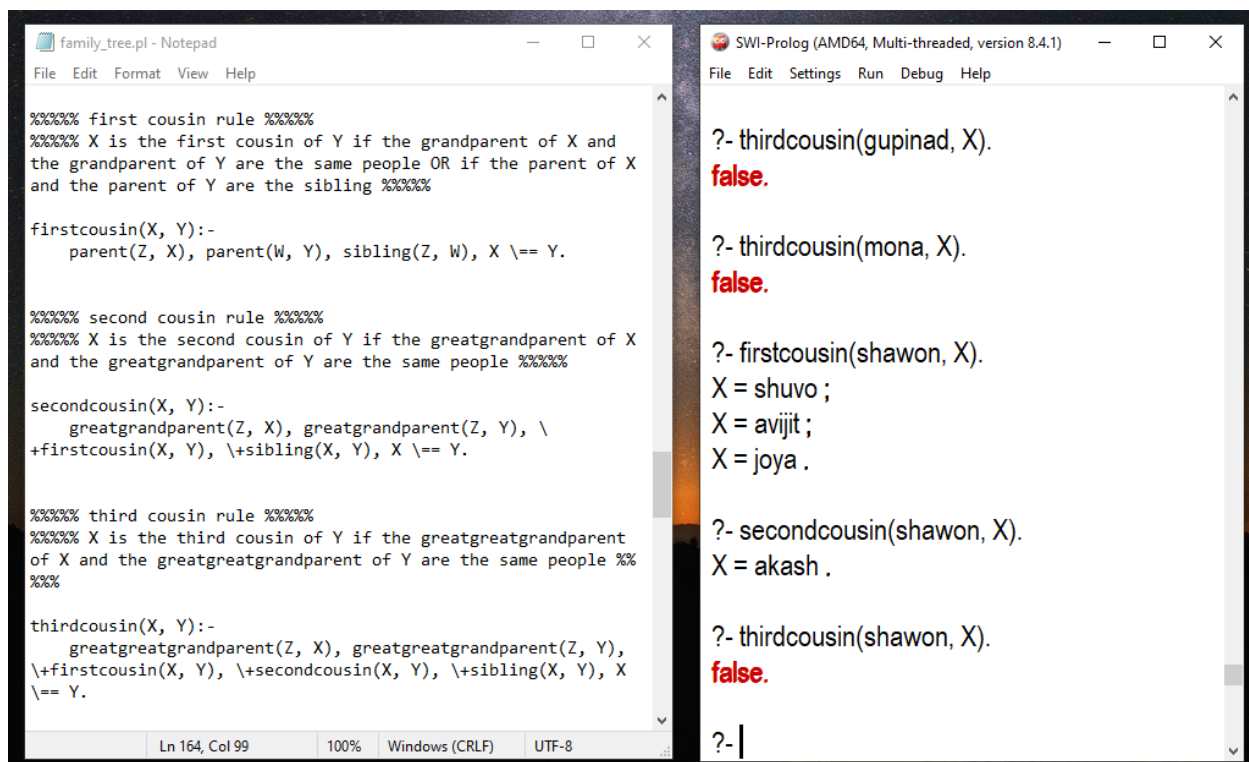
The screenshot shows two windows. The left window, titled 'family_tree.pl - Notepad', contains the following Prolog code:

```
%%% uncle rule %%%  
  
uncle(X, Y):-  
    father(Z, Y), father(A, Z), father(A, X), Z  
    \== X.  
  
Ln 145, Col 1      100%  Windows (CRLF)  UTF-8
```

The right window, titled 'SWI-Prolog (AMD64, Multi-threaded, version 8.4.1)', shows the query and its result:

```
?- uncle(ajit, shawon).  
true.  
  
?- |
```

In the below screenshot, here is the sample input output for first cousin, second cousin, and third cousin for a individual person.



The screenshot shows two windows. The left window, titled 'family_tree.pl - Notepad', contains the following Prolog code:

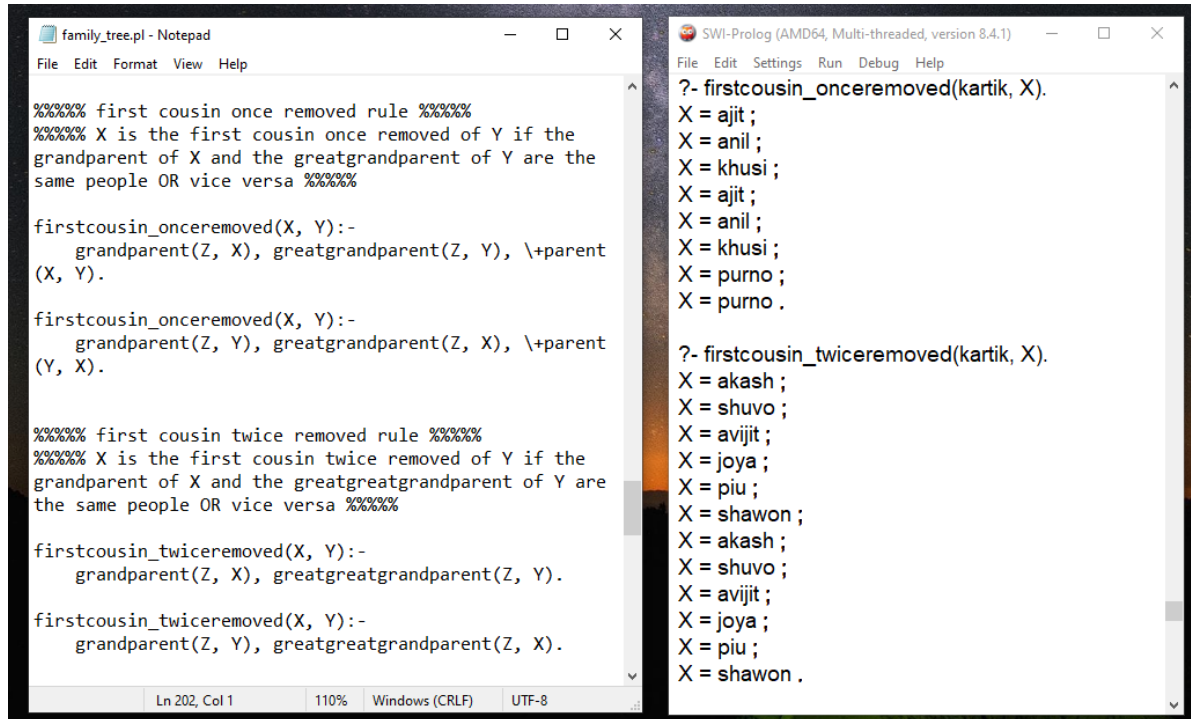
```
%%% first cousin rule %%%  
%%% X is the first cousin of Y if the grandparent of X and  
%%% the grandparent of Y are the same people OR if the parent of X  
%%% and the parent of Y are the sibling %%%  
  
firstcousin(X, Y):-  
    parent(Z, X), parent(W, Y), sibling(Z, W), X \== Y.  
  
%%% second cousin rule %%%  
%%% X is the second cousin of Y if the greatgrandparent of X  
%%% and the greatgrandparent of Y are the same people %%%  
  
secondcousin(X, Y):-  
    greatgrandparent(Z, X), greatgrandparent(Z, Y), \  
    +firstcousin(X, Y), \+sibling(X, Y), X \== Y.  
  
%%% third cousin rule %%%  
%%% X is the third cousin of Y if the greatgreatgrandparent  
%%% of X and the greatgreatgrandparent of Y are the same people %%%  
%%%  
  
thirdcousin(X, Y):-  
    greatgreatgrandparent(Z, X), greatgreatgrandparent(Z, Y),  
    \+firstcousin(X, Y), \+secondcousin(X, Y), \+sibling(X, Y), X  
    \== Y.  
  
Ln 164, Col 99      100%  Windows (CRLF)  UTF-8
```

The right window, titled 'SWI-Prolog (AMD64, Multi-threaded, version 8.4.1)', shows the queries and their results:

```
?- thirdcousin(gupinad, X).  
false.  
  
?- thirdcousin(mona, X).  
false.  
  
?- firstcousin(shawon, X).  
X = shuvo ;  
X = avijit ;  
X = joya .  
  
?- secondcousin(shawon, X).  
X = akash .  
  
?- thirdcousin(shawon, X).  
false.  
  
?- |
```

Name: Shawon Barman
Reg ID: 18201043

In the below screenshot, here is the sample input output for first cousin once removed, and first cousin twice removed for a individual person.



The screenshot shows two windows. The left window, titled 'family_tree.pl - Notepad', contains Prolog rules for first cousin relationships. The right window, titled 'SWI-Prolog (AMD64, Multi-threaded, version 8.4.1)', shows the results of queries for 'kartik'.

```
family_tree.pl - Notepad
File Edit Format View Help

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% first cousin once removed rule %%%%%%%%%
%%%%%%%% X is the first cousin once removed of Y if the
grandparent of X and the greatgrandparent of Y are the
same people OR vice versa %%%%%%%%%

firstcousin_onceremoved(X, Y):-
    grandparent(Z, X), greatgrandparent(Z, Y), \+parent
(X, Y).

firstcousin_onceremoved(X, Y):-
    grandparent(Z, Y), greatgrandparent(Z, X), \+parent
(Y, X).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% first cousin twice removed rule %%%%%%%%%
%%%%%%%% X is the first cousin twice removed of Y if the
grandparent of X and the greatgreatgrandparent of Y are
the same people OR vice versa %%%%%%%%%

firstcousin_twiceremoved(X, Y):-
    grandparent(Z, X), greatgreatgrandparent(Z, Y).

firstcousin_twiceremoved(X, Y):-
    grandparent(Z, Y), greatgreatgrandparent(Z, X).

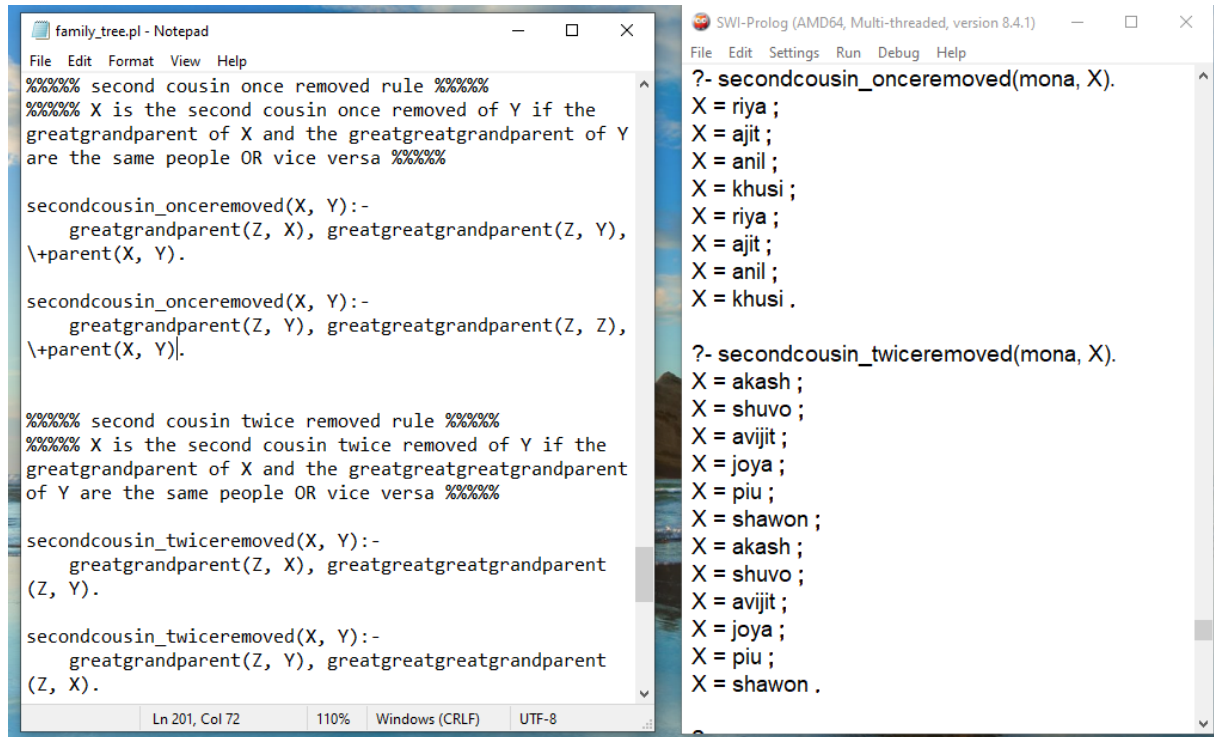
Ln 202, Col 1    110%    Windows (CRLF)    UTF-8

SWI-Prolog (AMD64, Multi-threaded, version 8.4.1)
File Edit Settings Run Debug Help

?- firstcousin_onceremoved(kartik, X).
X = ajit ;
X = anil ;
X = khushi ;
X = ajit ;
X = anil ;
X = khushi ;
X = purno ;
X = purno .

?- firstcousin_twiceremoved(kartik, X).
X = akash ;
X = shuvo ;
X = avijit ;
X = joya ;
X = piu ;
X = shawon ;
X = akash ;
X = shuvo ;
X = avijit ;
X = joya ;
X = piu ;
X = shawon .
```

In the below screenshot, here is the sample input output for second cousin once removed, and second cousin twice removed for a individual person.



The screenshot shows two windows. The left window, titled 'family_tree.pl - Notepad', contains Prolog rules for second cousin relationships. The right window, titled 'SWI-Prolog (AMD64, Multi-threaded, version 8.4.1)', shows the results of queries for 'mona'.

```
family_tree.pl - Notepad
File Edit Format View Help

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% second cousin once removed rule %%%%%%%%%
%%%%%%%% X is the second cousin once removed of Y if the
greatgrandparent of X and the greatgreatgrandparent of Y
are the same people OR vice versa %%%%%%%%%

secondcousin_onceremoved(X, Y):-
    greatgrandparent(Z, X), greatgreatgrandparent(Z, Y),
\+parent(X, Y).

secondcousin_onceremoved(X, Y):-
    greatgrandparent(Z, Y), greatgreatgrandparent(Z, Z),
\+parent(X, Y).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% second cousin twice removed rule %%%%%%%%%
%%%%%%%% X is the second cousin twice removed of Y if the
greatgrandparent of X and the greatgreatgreatgrandparent
of Y are the same people OR vice versa %%%%%%%%%

secondcousin_twiceremoved(X, Y):-
    greatgrandparent(Z, X), greatgreatgreatgrandparent
(Z, Y).

secondcousin_twiceremoved(X, Y):-
    greatgrandparent(Z, Y), greatgreatgreatgrandparent
(Z, X).

Ln 201, Col 72    110%    Windows (CRLF)    UTF-8

SWI-Prolog (AMD64, Multi-threaded, version 8.4.1)
File Edit Settings Run Debug Help

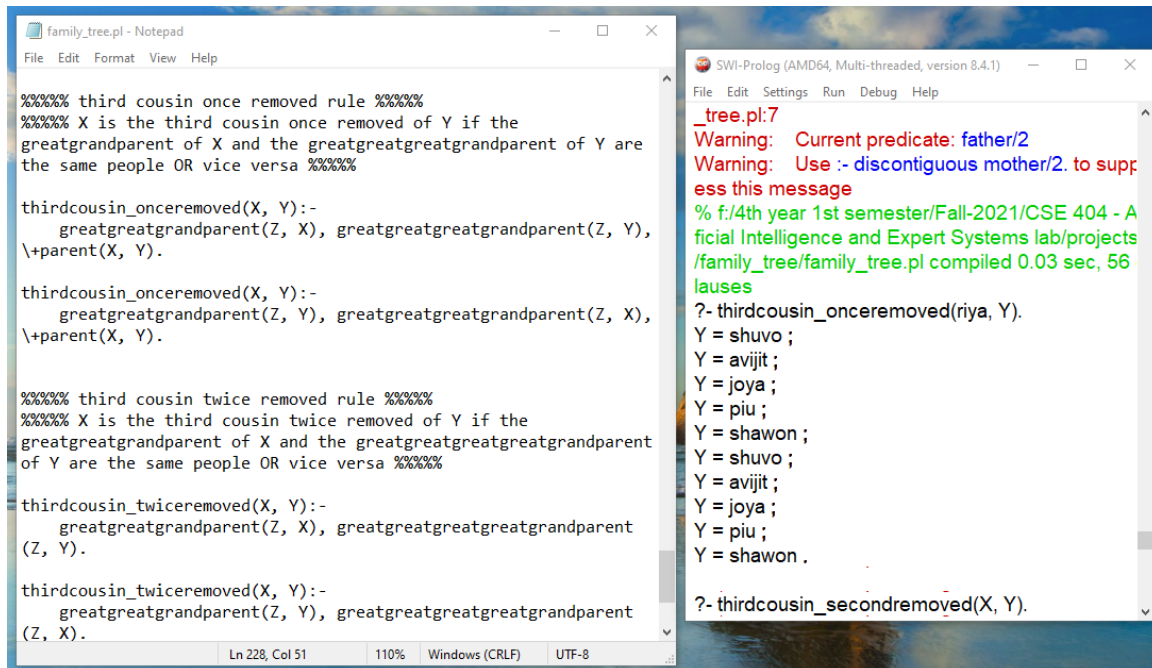
?- secondcousin_onceremoved(mona, X).
X = riya ;
X = ajit ;
X = anil ;
X = khushi ;
X = riya ;
X = ajit ;
X = anil ;
X = khushi .

?- secondcousin_twiceremoved(mona, X).
X = akash ;
X = shuvo ;
X = avijit ;
X = joya ;
X = piu ;
X = shawon ;
X = akash ;
X = shuvo ;
X = avijit ;
X = joya ;
X = piu ;
X = shawon .
```

Name: Shawon Barman

Reg ID: 18201043

In the below screenshot, here is the sample input output for third cousin once removed, and third cousin twice removed for a individual person.



The screenshot shows two windows. The left window is a Notepad editor titled 'family_tree.pl' containing Prolog code. The right window is the SWI-Prolog IDE (version 8.4.1) showing the execution of the code.

```
family_tree.pl - Notepad
File Edit Format View Help

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% third cousin once removed rule %%%%%%%%%
%%%%%%%% X is the third cousin once removed of Y if the
%%%%%%%% greatgrandparent of X and the greatgreatgreatgrandparent of Y are
%%%%%%%% the same people OR vice versa %%%%%%%%%

thirdcousin_onceremoved(X, Y):-
    greatgreatgrandparent(Z, X), greatgreatgreatgrandparent(Z, Y),
    \+parent(X, Y).

thirdcousin_onceremoved(X, Y):-
    greatgreatgrandparent(Z, Y), greatgreatgreatgrandparent(Z, X),
    \+parent(X, Y).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% third cousin twice removed rule %%%%%%%%%
%%%%%%%% X is the third cousin twice removed of Y if the
%%%%%%%% greatgreatgrandparent of X and the greatgreatgreatgreatgrandparent
%%%%%%%% of Y are the same people OR vice versa %%%%%%%%%

thirdcousin_twiceremoved(X, Y):-
    greatgreatgrandparent(Z, X), greatgreatgreatgreatgrandparent
    (Z, Y).

thirdcousin_twiceremoved(X, Y):-
    greatgreatgrandparent(Z, Y), greatgreatgreatgreatgrandparent
    (Z, X).

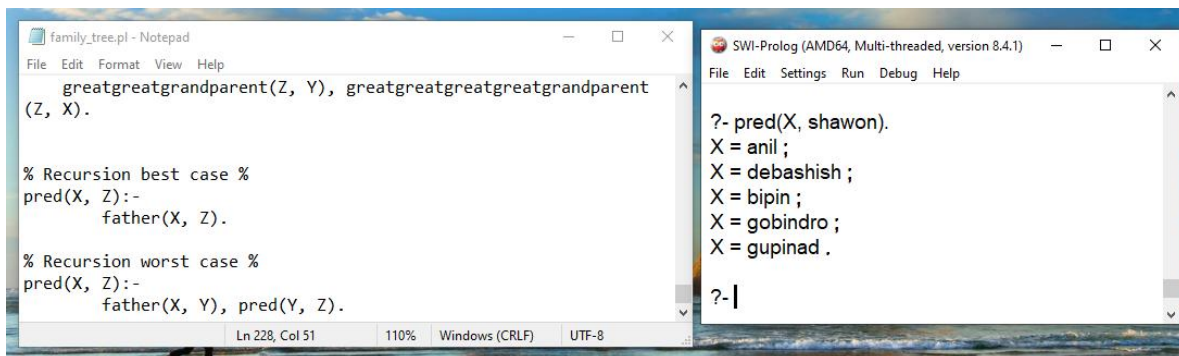
Ln 228, Col 51    110%    Windows (CRLF)    UTF-8
```

```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.1)
File Edit Settings Run Debug Help

_tree.pl:7
Warning: Current predicate: father/2
Warning: Use :- discontinuous mother/2. to suppress this message
% f:/4th year 1st semester/Fall-2021/CSE 404 - Artificial Intelligence and Expert Systems lab/projects
/family_tree/family_tree.pl compiled 0.03 sec, 56
lauses
?- thirdcousin_onceremoved(riya, Y).
Y = shuvo ;
Y = avijit ;
Y = joya ;
Y = piu ;
Y = shawon ;
Y = shuvo ;
Y = avijit ;
Y = joya ;
Y = piu ;
Y = shawon .

?- thirdcousin_twiceremoved(X, Y).
```

Who is the predecessor of shawon. In the below screenshot, here is the sample input and output:



The screenshot shows two windows. The left window is a Notepad editor titled 'family_tree.pl' containing Prolog code. The right window is the SWI-Prolog IDE (version 8.4.1) showing the execution of the code.

```
family_tree.pl - Notepad
File Edit Format View Help

greatgreatgrandparent(Z, Y), greatgreatgreatgreatgrandparent
(Z, X).

% Recursion best case %
pred(X, Z):-
    father(X, Z).

% Recursion worst case %
pred(X, Z):-
    father(X, Y), pred(Y, Z).

Ln 228, Col 51    110%    Windows (CRLF)    UTF-8
```

```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.1)
File Edit Settings Run Debug Help

?- pred(X, shawon).
X = anil ;
X = debashish ;
X = bipin ;
X = gobindro ;
X = gupinad .

?- |
```

Conclusion:

I've faced some minor difficulties during completing this assignment. SWI-Prolog was showing some errors. But after some troubleshooting I was able to fix all the errors of SWI-Prolog. <https://swi-prolog.discourse.group/> has some amazing solution which helps me a lot during troubleshooting