AI Lab Assignment

```
%Prolog Assignment 1: Build Fact, Rule, goal for family relationships
and arithmetic operations using Prolog.

%Submitted by: Sanskar Sharma

%PRN: 0120180381

%Roll number: 090 TY

%*******************************Assignment*****************************

%Facts for the family

male(neev).

male(raju).

male(sans).

male(zulu).

male(andrew).

male(lee).

female(anna).

female(taylor).

female(jessy).

female(kristen).


parent_of(neev,jessy).

parent_of(neev,kristen).

parent_of(anna, jessy).

parent_of(anna, kristen).

parent_of(raju,james).

parent_of(taylor, james).

parent_of(jessy, andrew).

parent_of(sans, andrew).

parent_of(kristen, lee).

parent_of(zulu, lee).


%Rules for the family

father_of(X,Y):- male(X),

    parent_of(X,Y).
```

```
mother_of(X,Y):- female(X),
    parent_of(X,Y).

grandfather_of(X,Y):- male(X),
    parent_of(X,Z),
    parent_of(Z,Y).

grandmother_of(X,Y):- female(X),
    parent_of(X,Z),
    parent_of(Z,Y).

sister_of(X,Y):- %(X,Y or Y,X)%
    female(X),
    father_of(F, Y), father_of(F,X),X \= Y.

sister_of(X,Y):- female(X),
    mother_of(M, Y), mother_of(M,X),X \= Y.

aunt_of(X,Y):- female(X),
    parent_of(Z,Y), sister_of(Z,X),!.

brother_of(X,Y):- %(X,Y or Y,X)%
    male(X),
    father_of(F, Y), father_of(F,X),X \= Y.

brother_of(X,Y):- male(X),
    mother_of(M, Y), mother_of(M,X),X \= Y.

uncle_of(X,Y):-
    parent_of(Z,Y), brother_of(Z,X).
```

% Goals and Queries.(Output of family relations for given facts and rules)

```
/*
% c:/Users/hp/Documents/Prolog/family.pl compiled 0.00 sec, 30 clauses
?-  male(neev).
```

```
true.

?- male(anna).
false.

?- female(kristen).
true.

?- male(X).
X = neev ;
X = raju ;
X = sans ;
X = zulu ;
X = andrew ;
X = lee.

?- father_of(X,kristen).
X = neev ;
false.

?- mother_of(X,lee).
X = kristen.

?- parent_of(X,lee).
X = kristen ;
X = zulu.

?- grandmother_of(X,andrew).
X = anna ;
false.

?- grandfather_of(X,andrew).
X = neev ;
false.

?- sister_of(kristen,X).
```

```
X = jessy ;
X = jessy ;
false.

?- brother_of(X,Y).
false.

?- aunt_of(jessy,X).
X = lee.

?- uncle_of(sans,Y).
false.
*/

%Arithmetic operations

%Find the N'th element of a list.
ele_at(X,[X|_],1).
ele_at(X,[_|L],N) :- N > 1, N1 is N - 1, ele_at(X,L,N1).
/*output:
?- element_at(X,[a,b,c,d,e],3).
X = c.
*/

%Factorial of a number using recurssion
fact(0,1).
fact(N,F) :-
   N>0,
   N1 is N-1,
   fact(N1,F1),
   F is N * F1.
/*Output
?-  fact(6,F).
F = 720 ;
false.
```

```
?- fact(0,F).
F = 1.
*/

%Tower of Hanoi Problem using recurssion
mov(1,X,Y,_) :-
    write('Move topmost disk from '),
    write(X),
    write(' to '),
    write(Y),
    nl.
mov(N,X,Y,Z) :-
    N>1,
    M is N-1,
    mov(M,X,Z,Y),
    mov(1,X,Y,_),
    mov(M,Z,Y,X).
/*output:
?- mov(5,a,b,c).
Move topmost disk from a to b
Move topmost disk from a to c
Move topmost disk from b to c
Move topmost disk from a to b
Move topmost disk from c to a
Move topmost disk from c to b
Move topmost disk from a to b
Move topmost disk from a to c
Move topmost disk from b to c
Move topmost disk from b to a
Move topmost disk from c to a
Move topmost disk from b to c
Move topmost disk from a to b
Move topmost disk from a to c
Move topmost disk from b to c
Move topmost disk from a to b
Move topmost disk from c to a
```

```
Move topmost disk from c to b
Move topmost disk from a to b
Move topmost disk from c to a
Move topmost disk from b to c
Move topmost disk from b to a
Move topmost disk from c to a
Move topmost disk from c to b
Move topmost disk from a to b
Move topmost disk from a to c
Move topmost disk from b to c
Move topmost disk from a to b
Move topmost disk from c to a
Move topmost disk from c to b
Move topmost disk from a to b
true
*/
```