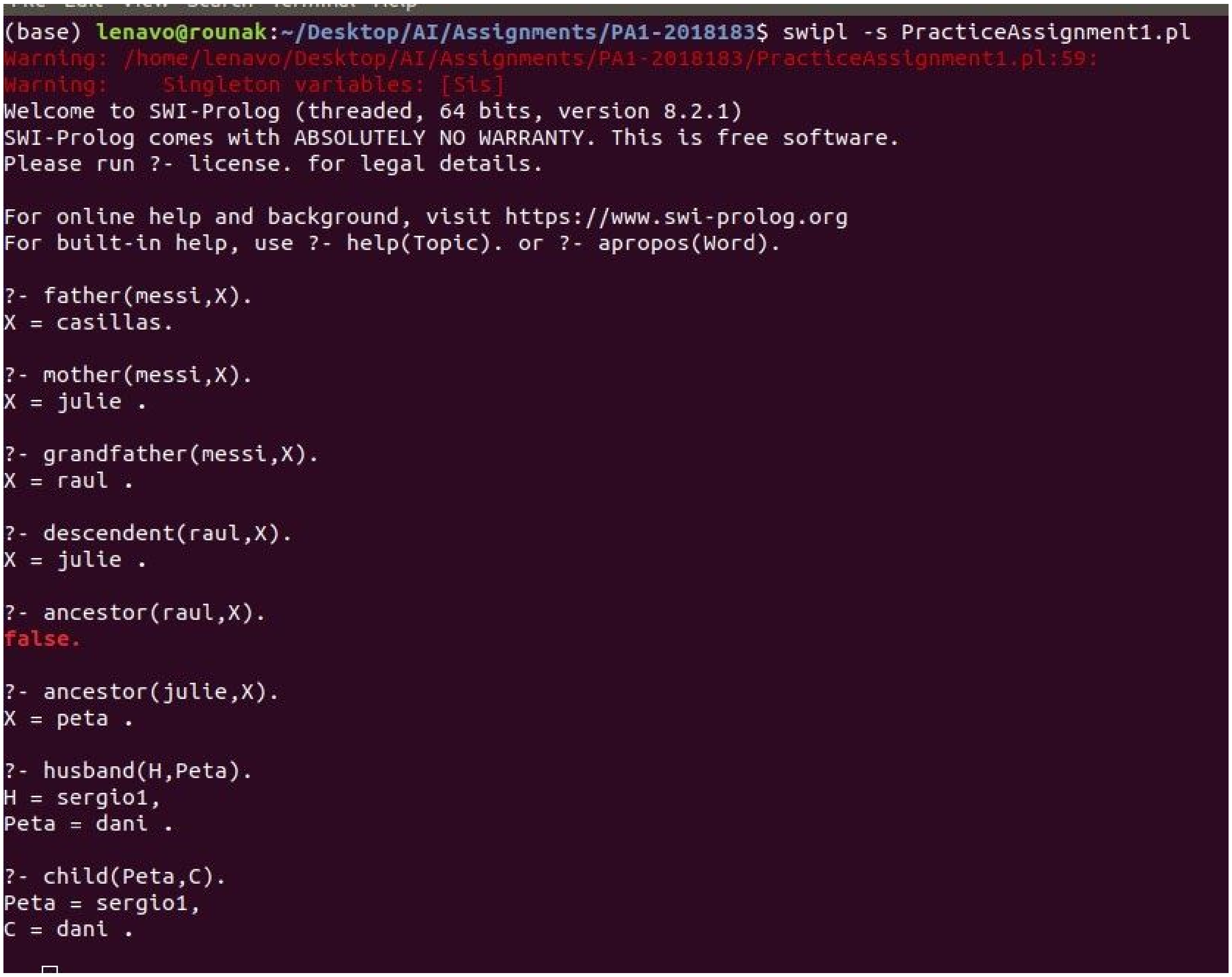
ARTIFICIAL INTELLIGENCE

PRACTISE ASSIGNMENT 1

SCREENSHOT AND LISTING



```
(base) lenavo@rounak:~/Desktop/AI/Assignments/PA1-2018183$ swipl -s PracticeAssignment1.pl
Warning: /home/lenavo/Desktop/AI/Assignments/PA1-2018183/PracticeAssignment1.pl:59:
Warning:    Singleton variables: [Sis]
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.1)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- father(messi,X).
X = casillas.

?- mother(messi,X).
X = julie .

?- grandfather(messi,X).
X = raul .

?- descendent(raul,X).
X = julie .

?- ancestor(raul,X).
false.

?- ancestor(julie,X).
X = peta .

?- husband(H,Peta).
H = sergio1,
Peta = dani .

?- child(Peta,C).
Peta = sergio1,
C = dani .
```

Listing of the program -

% Defining few male predicates

male(sergio1).

male(cristiano1).

```
male(cristiano2).

male(sergio2).

male(messi).

male(raul).

male(casillas).

% Defining few female predicates

female(alison).

female(peta).

female(julie).

female(dani).

female(faye).

% Defining few parent predicates parent(child,parent)

parent(sergio1,dani).

parent(peta,sergio1).

parent(cristiano2,cristiano1).

parent(alison,cristiano1).

parent(sergio2,cristiano1).

parent(julie,peta).

parent(messi,julie).

parent(messi,casillas).

parent(alison,faye).
parent(cristiano2,faye).
parent(sergio2,faye).
parent(julie,raul).
parent(peta,dani).
parent(cristiano1,dani).
% married (A,B) - A is married to B
married(sergio1,dani).
married(dani,sergio1).
married(cristiano1,faye).
married(faye,cristiano1).
married(peta,raul).
married(raul,peta).
married(julie,casillas).
married(casillas,julie).

% DEFINING RELATIONS
% Dad - 1) Male 2) Childs parent
father(Child,Dad) :- male(Dad),parent(Child,Dad).
% Mom - 1) Female 2) Childs parent
mother(Child,Mom) :- female(Mom),parent(Child,Mom).
% Bro - 1) Male 2) Same parents as the sibling
```

```
brother(Sibling,Bro) :- male(Bro),father(Sibling,Father),father(Bro,Father),Bro\=Sibling,mother(Sibling,Mother),mother(Sis,Mother).

% Sis - 1) Female 2) Same parents as the sibling

sister(Sibling,Sis) :- female(Sis),father(Sibling,Father),father(Sis,Father),Sis\=Sibling,mother(Sibling,Mother),mother(Sis,Mother).



% Aunt -

% Condition 1 - 1) Female 2) Sister of ones father or mother

% Condition 2 - 1) Female 2) Wife of ones uncle

aunt(Kid,Auntie) :- female(Auntie),parent(Kid,Parent),sister(Parent,Auntie).

aunt(Kid,Auntie) :- female(Auntie),parent(Kid,Person),brother(Person,Brother),married(Auntie,Brother).

% Uncle -

% Condition 1 - 1) Male 2) Brother of ones father or mother

% Condition 2 - 1) Male 2) Husband of ones aunt

uncle(Kid,Uncle) :- male(Uncle),parent(Kid,Parent),brother(Parent,Uncle).

uncle(Kid,Uncle) :- male(Uncle),parent(Kid,Person),sister(Person,Sister),married(Uncle,Sister).

% Grandmother - 1) Female 2) Parent of granchilds parent

grandmother(Grandchild,Grandmother) :- female(Grandmother),parent(Grandchild,Parent),parent(Parent,Grandmother).

% Grandfather - 1) Male 2) Parent of grandchilds parent
```

```
grandfather(Grandchild,Grandfather) :-
male(Grandfather),parent(Grandchild,Parent),parent(Parent,Grandfather).

%Brother-In-Law -

%Condition 1 - 1) Male 2) Brother of ones spouse

%Condition 2 - 1) Male 2) Husbands of ones sister

%Condition 3 - 1) Male 2) Husband of the sister of ones spouse

brother_in_law(SiblingInLaw,BInLaw) :-
married(SiblingInLaw,Person),brother(Person,BInLaw).

brother_in_law(SiblingInLaw,BInLaw) :- sister(SiblingInLaw,Sister),married(Sister,BInLaw).

brother_in_law(SiblingInLaw,BInLaw) :-
married(SiblingInLaw,Person),sister(Person,Sister),married(Sister,BInLaw).

% Sister-In-Law -

%Condition 1 - 1) Female 2) Sister of ones spouse

%Condition 2 - 1) Female 2) Wife of ones brother

%Condition 3 - 1) Female 2) Wife of brother of ones spouse

sister_in_law(SiblingInLaw,SInLaw) :- married(SiblingInLaw,Person),sister(Person,SInLaw).

sister_in_law(SiblingInLaw,SInLaw) :-
brother(SiblingInLaw,Brother),married(Brother,SInLaw).

sister_in_law(SiblingInLaw,SInLaw) :-
married(SiblingInLaw,Person),brother(Person,Brother),married(Brother,SInLaw).

% Mother-In-Law -

% 1) Female 2) Son is married to child-in-law 3) Daughter is married to child-in-law.
```

```
mother_in_law(ChildInLaw,MInLaw) :-
female(MInLaw),married(ChildInLaw,Child),parent(Child,MInLaw).

% Father-In-Law -

% 1) Male 2) Son is married to child-in-law 3) Daughter is married to child-in-law.

father_in_law(ChildInLaw,FInLaw) :-
male(FInLaw),married(ChildInLaw,Child),parent(Child,FInLaw).

% Ancestor - 1) Parent of the Person 2) Parent of the persons parent and so on.

ancestor(Person,Ancestor) :- parent(Person,Ancestor).

ancestor(Person,Ancestor) :- parent(Person,Parent),ancestor(Parent,Ancestor).

% Descendent - 1) Child of the Person 2) Child of the Parent whos parent is Person and so on.

descendent(Person,Descendent) :- parent(Descendent,Person).

descendent(Person,Descendent) :-
parent(Descendent,Someone),descendent(Person,Someone).

% Son - 1) Male 2) Child of a Person

son(Son,Parent) :- male(Son),parent(Son,Parent).

% Daughter - 1) Female 2) Child of a Person

daughter(Daughter,Parent) :- female(Daughter),parent(Daughter,Parent).

% Sibling -

%1) Brother of Person and male

%2) Sister of Person and female

sibling(Test,Person) :- female(Test),sister(Person,Test).

sibling(Test,Person) :- male(Test),brother(Person,Test).
```

```
% Child - parent of Child is Parent

child(Child,Parent) :- parent(Child,Parent).

% Husband - 1) Male 2) Married to a Person

husband(Husband,Person) :- male(Husband),married(Husband,Person).

% Wife - 1) Female 2) Married to a Person

wife(Wife,Person) :- female(Wife),married(Wife,Person).

% First Cousin - 1) Child of parents siblings

first_cousin(Test,Cousin) :-
parent(Test,Parent),sibling(Parent,ParentSibling),child(Cousin,ParentSibling).

% Second Cousin - 1) Child of parents first cousins

second_cousin(Test,Cousin) :-
parent(Test,Parent),first_cousin(Parent,First_Cousin),child(Cousin,First_Cousin).

% Grandaunt - parents aunt

grandaunt(GrandAunt,Person) :- parent(Person,Parent),aunt(Parent,GrandAunt).

% Granduncle- parents uncle

granduncle(GrandUncle,Person) :- parent(Person,Parent),uncle(Parent,GrandUncle).
```