ASSIGNMENT – 2

Load the following facts into familytree.pl , consult the prolog file and answer the given questions

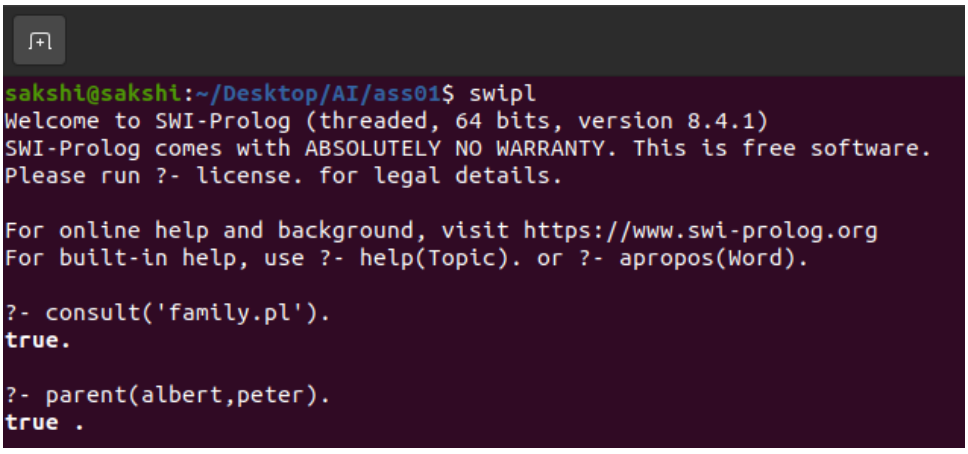
```
% Program: family.pl

parent(albert, jim).
parent(albert, peter).
parent(jim, brian).
parent(john, darren).
parent(peter, lee).
parent(peter, sandra).
parent(peter, james).
parent(peter, kate).
parent(peter, kyle).
parent(brian, jenny).
parent(irene, jim).
parent(irene, peter).
parent(pat, brian).
parent(pat, darren).
parent(amanda, jenny).

% female(Person)
%
female(irene).
female(pat).
female(lee).
female(sandra).
female(jenny).
female(amanda).
female(kate).

% male(Person)
%
male(albert).
male(jim).
male(peter).
male(brian).
```

male(john).
male(darren).
male(james).
male(kyle).

% yearOfBirth(Person, Year).
%
yearOfBirth(irene, 1923).
yearOfBirth(pat, 1954).
yearOfBirth(lee, 1970).
yearOfBirth(sandra, 1973).
yearOfBirth(jenny, 2004).
yearOfBirth(amanda, 1979).
yearOfBirth(albert, 1926).
yearOfBirth(jim, 1949).
yearOfBirth(peter, 1945).
yearOfBirth(brian, 1974).
yearOfBirth(john, 1955).
yearOfBirth(darren, 1976).
yearOfBirth(james, 1969).
yearOfBirth(kate, 1975).
yearOfBirth(kyle, 1976).

Use SWI – Prolog for answering the following questions (load the rules in the file familytree.pl):

1. Is Albert a parent of Peter?



2. Who is the child of Jim?

```
?- parent(jim,X).
X = brian.
```

3. Who are the parents of Brian?

```
?- parent(X,brian).
X = jim ;
X = pat.

?-
```

4. Is Irene a grandparent of Brian?

```
% grandparent(Gparent,Child)
grand_parent(X,Y) :-
    parent(X,Z),
    parent(Z,Y).
```

```
sakshi@sakshi:~/Desktop/AI/ass01$ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.1)
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).

?- consult('family.pl').
true.

?- grand_parent(irene,brian).
true .
```

5. Find all the grandchildren of Irene

```
?- grand_parent(irene,X).
X = brian ;
X = lee ;
X = sandra ;
X = james ;
X = kate ;
X = kyle.
```

6. Now add the following rule to familytree.pl and re-consult:

older(Person1, Person2) :- yearOfBirth(Person1, Year1), yearOfBirth(Person2, Year2), Year2 > Year1.

```
?- consult('family.pl').
true.
```

7. Who is older than Pat?

```
?- older(X,pat).
X = irene ;
X = albert ;
X = jim ;
X = peter ;
false.
```

8. Who is younger than Darren?

```
?- older(darren,X).
X = jenny ;
X = amanda ;
false.

?-
```

9. List the siblings of Sandra.

```
% sibling
sibling(X,Y) :-
    parent(Z,X),
    parent(Z,Y),
    X \= Y.
```

```
?- sibling(sandra,X).
X = lee ;
X = james ;
X = kate ;
X = kyle.
```

10. Who is the older brother of Sandra?

```
% elder brother
older_brother(X,Y) :-
    male(X),
    sibling(X,Y),
    older(X,Y).
```

```
?- older_brother(X,sandra).
X = james ;
false.
```

11. Find the predecessors of Kyle.

```
%
```

```
predecessor(X,Y):-
    parent(X,Z),
    predecessor(Z,Y).
predecessor(X,Y):-
    parent(X,Y).
```

```
?- predecessor(X,kyle).
X = albert ;
X = irene ;
X = peter.

?-
```

12. Does Kate have a sister?

```
% sister
has_sister(X,Y) :-
    female(X),
    parent(Z,X),
    parent(Z,Y),
    X \= Y.
```

```
?- has_sister(X,kate).
X = lee ;
X = sandra ;
false.
```

13. How many females and males are there in the knowledge base?

```
% count rule
person(X) :- female(X).
person(Y) :- male(Y).
```

```
?- aggregate_all(count,person(X),Count).
Count = 15.
```