Practical – 1

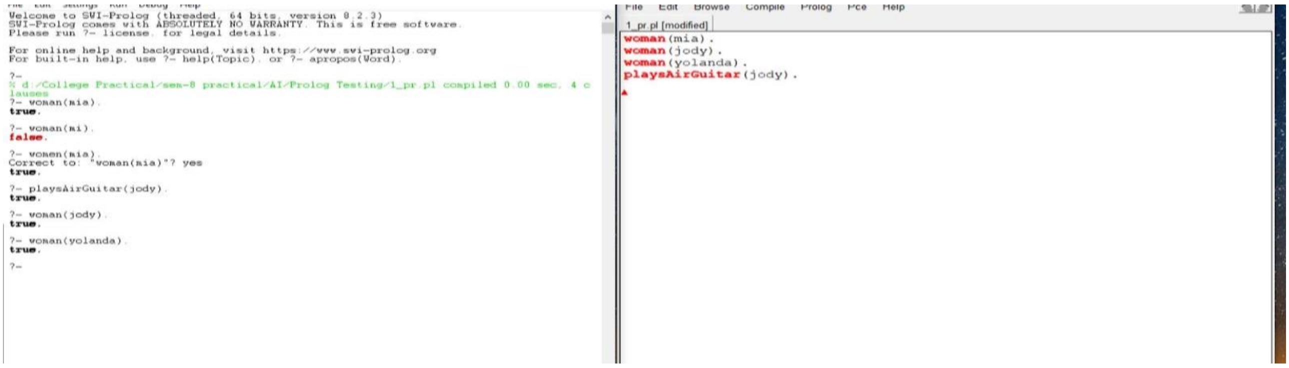
AIM 1:

A) Write prolog program to implement different kinds of knowledge bases.

1) Knowledge Base 1:

Program:

```
woman(mia).
woman(jody).
woman(yolanda).
playsAirGuitar(jody).
```



```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
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).

?-
% d:/College Practical/sem-8 practical/AI/Prolog Testing/1_pr.pl compiled 0.00 sec, 4 c
lauses
?- woman(mia).
true.

?- woman(mi).
false.

?- women(mia).
Correct to: "woman(mia)"? yes
true.

?- playsAirGuitar(jody).
true.

?- woman(jody).
true.

?- woman(yolanda).
true.

?-
```

```
woman(mia).
woman(jody).
woman(yolanda).
playsAirGuitar(jody).
```

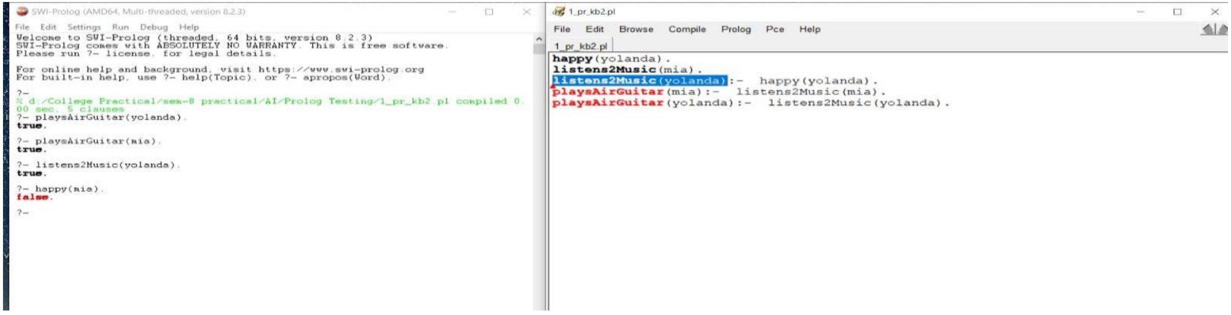
File Edit Settings Run Debug Help
File Edit Browse Compile Prolog Pce Help
1_pr.pl [modified]

Fig 1.1

2) Knowledge Base 2:

Program:

```
happy(yolanda). listens2Music(mia).
listens2Music(yolanda):-  happy(yolanda).
playsAirGuitar(mia):-  listens2Music(mia).
playsAirGuitar(yolanda):-  listens2Music(yolanda).
```



```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
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).

?-
% d:/College Practical/sem-8 practical/AI/Prolog Testing/1_pr_kb2.pl compiled 0.
00 sec, 5 clauses
?- playsAirGuitar(yolanda).
true.

?- playsAirGuitar(mia).
true.

?- listens2Music(yolanda).
true.

?- happy(mia).
false.

?-
```

```
happy(yolanda).
listens2Music(mia).
listens2Music(yolanda):-  happy(yolanda).
playsAirGuitar(mia):-  listens2Music(mia).
playsAirGuitar(yolanda):-  listens2Music(yolanda).
```

SWI-Prolog (AMD64, Multi-threaded, version 8.2.3)
File Edit Settings Run Debug Help
1_pr_kb2.pl
File Edit Browse Compile Prolog Pce Help
1_pr_kb2.pl

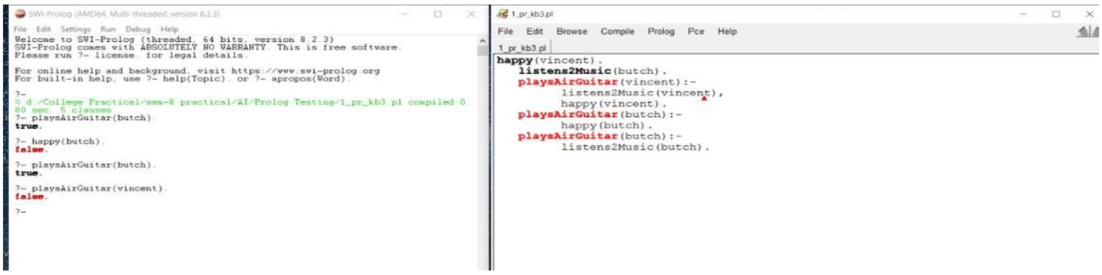
Fig 1.2

3) Knowledge Base 3:

Program:

```
happy(vincent).
listen2Music(vincent).
```

```
playsAirGuitar(vincent):- listen2Music(vincent), happy(vincent).
playsAirGuitar(butch):- happy(butch).
playsAirGuitar(butch):- listen2Music(butch).
```


SWI-Prolog (AMD64, Multi-threaded, version 8.2.3)
File Edit Settings Run Debug Help
1_pr_kb3.pl
File Edit Browse Compile Prolog Pce Help
1_pr_kb3.pl

```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
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).

?-
% d:/College Practical/sem-8 practical/AI/Prolog Testing/1_pr_kb3.pl compiled 0.
00 sec, 5 clauses
?- playsAirGuitar(butch).
true.

?- happy(butch).
false.

?- playsAirGuitar(butch).
true.

?- playsAirGuitar(vincent).
false.

?-
```

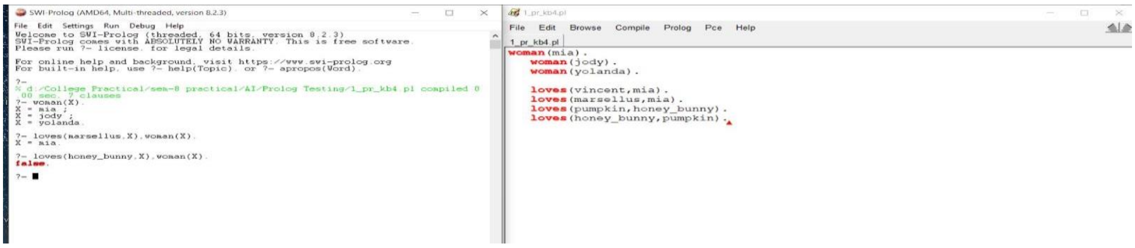
```
happy(vincent).
   listens2Music(butch).
   playsAirGuitar(vincent):-
        listens2Music(vincent),
        happy(vincent).
   playsAirGuitar(butch):-
        happy(butch).
   playsAirGuitar(butch):-
        listens2Music(butch).
```

Fig 1.3

4) Knowledge Base 4:

Program:

```
woman(mia).
woman(jody).
woman(yolanda).
loves(vincent,mia).
loves(marsellus,mia).
loves(pumpkin,honey_bunny).
loves(honey_bunny,pumpkin).
```


SWI-Prolog (AMD64, Multi-threaded, version 8.2.3)
File Edit Settings Run Debug Help
1_pr_kb4.pl
File Edit Browse Compile Prolog Pce Help
1_pr_kb4.pl

```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
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).

?-
% d:/College Practical/sem-8 practical/AI/Prolog Testing/1_pr_kb4.pl compiled 0
.00 sec, 7 clauses
?- woman(X).
X = mia ;
X = jody ;
X = yolanda.

?- loves(marsellus,X),woman(X).
X = mia.

?- loves(honey_bunny,X),woman(X).
false.

?- ■
```

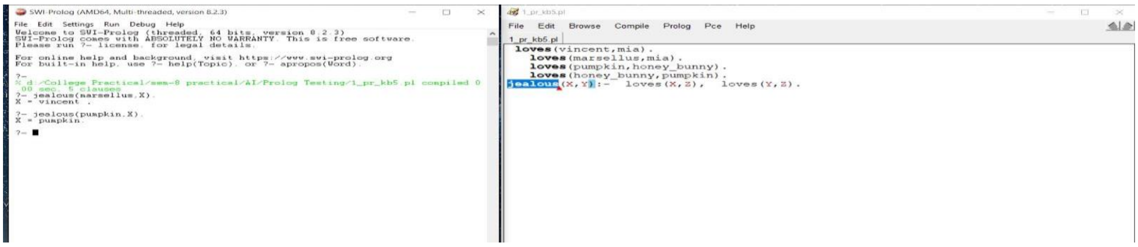
```
woman(mia).
   woman(jody).
   woman(yolanda).

   loves(vincent,mia).
   loves(marsellus,mia).
   loves(pumpkin,honey_bunny).
   loves(honey_bunny,pumpkin).
```

Fig 1.4

5) Knowledge Base 5:

Program:

```
loves(vincent,mia).
loves(marsellus,mia).
loves(pumpkin,honey_bunny).
loves(honey_bunny,pumpkin).
jealous(X,Y):-  loves(X,Z),  loves(Y,Z).
```


SWI-Prolog (AMD64, Multi-threaded, version 8.2.3)
File Edit Settings Run Debug Help
1_pr_kb5.pl
File Edit Browse Compile Prolog Pce Help
1_pr_kb5.pl

```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
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).

?-
% d:/College Practical/sem-8 practical/AI/Prolog Testing/1_pr_kb5.pl compiled 0
.00 sec, 5 clauses
?- jealous(marsellus,X).
X = vincent .

?- jealous(pumpkin,X).
X = pumpkin.

?- ■
```

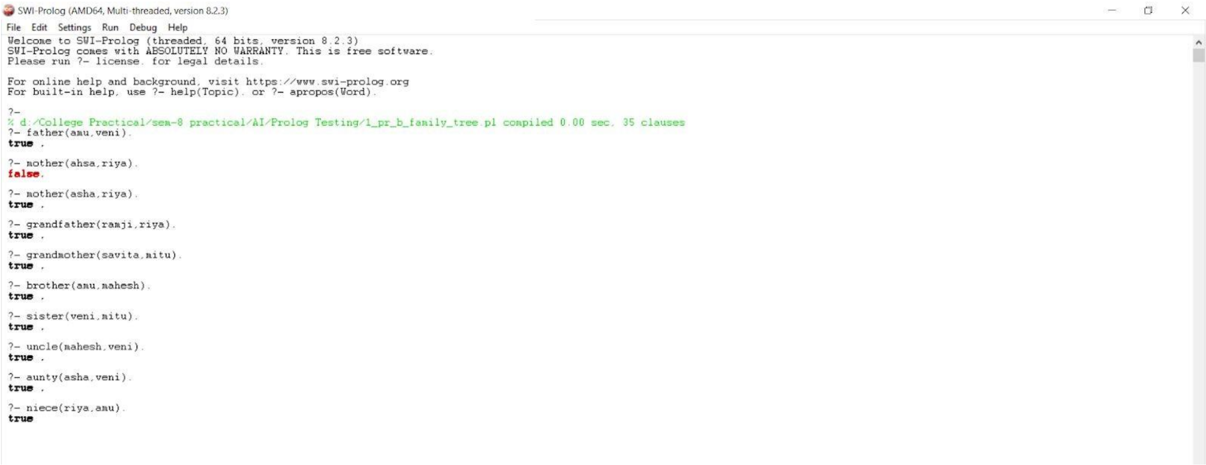
```
  loves(vincent,mia).
   loves(marsellus,mia).
   loves(pumpkin,honey_bunny).
   loves(honey_bunny,pumpkin).
jealous(X,Y):-  loves(X,Z),  loves(Y,Z).
```

Fig 1.5

B) Write a program which contains three predicates: male, female, parent. Make rules for following family relations: father, mother, grandfather, grandmother, brother, sister, uncle, aunt, nephew and niece.

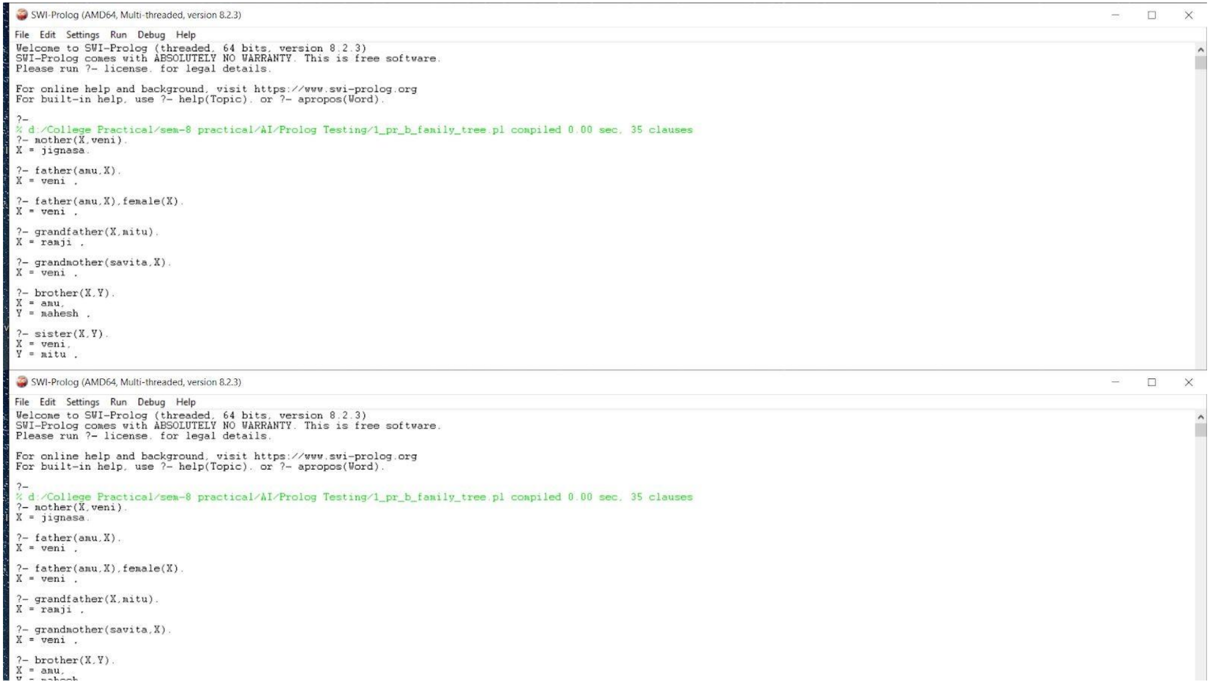
Program:

```
male(ramji).
male(amu).
male(mahesh).
female(savita).
female(jignasa).
female(asha).
female(veni).
female(mitu).
female(riya).
female(mahek).
parent(amu,veni).
parent(jignasa,veni).
parent(amu,mitu).
parent(jignasa,mitu).
parent(mahesh,riya).
parent(asha,riya).
parent(mahesh,mahek).
parent(asha,mahek).
parent(ramji,amu).
parent(savita,amu).
parent(ramji,mahesh).
parent(savita,mahesh).

son(X,Y):-male(X),parent(Y,X).
daughter(X,Y):-female(X),parent(Y,X).
mother(X,Y):-parent(X,Y),female(X).
father(X,Y):-parent(X,Y),male(X).
sister(X,Y):-parent(Z,X),parent(Z,Y),female(X),X\==Y.
brother(X,Y):-parent(Z,X),parent(Z,Y),male(X),X\==Y. grandfather(X,Y):-
parent(X,Z),parent(Z,Y),male(X),male(Z). grandmother(X,Y):-
parent(X,Z),parent(Z,Y),female(X),male(Z). uncle(X,Y):-parent(Z,Y),brother(Z,X),male(X).
wife(X,Y):-female(X),male(Y),parent(X,Z),parent(Y,Z).
aunty(X,Y):-female(X),parent(Z,Y),brother(P,Z),wife(X,P).
nephew(X,Y):-male(X),son(X,Z),brother(Z,Y).
niece(X,Y):-female(X),daughter(X,Z),brother(Z,Y).
```



```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
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).

?-
% d:/College Practical/sem-8 practical/AI/Prolog Testing/1_pr_b_family_tree.pl compiled 0.00 sec, 35 clauses
?- father(amu,veni).
true .

?- mother(ahsa,riya).
false.

?- mother(asha,riya).
true .

?- grandfather(ramji,riya).
true .

?- grandmother(savita,mitu).
true .

?- brother(amu,mahesh).
true .

?- sister(veni,mitu).
true .

?- uncle(mahesh,veni).
true .

?- aunty(asha,veni).
true .

?- niece(riya,amu).
true
```

Fig 1.6 Family tree



```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
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).

?-
% d:/College Practical/sem-8 practical/AI/Prolog Testing/1_pr_b_family_tree.pl compiled 0.00 sec, 35 clauses
?- mother(X,veni).
X = jignasa.

?- father(amu,X).
X = veni .

?- father(amu,X),female(X).
X = veni .

?- grandfather(X,mitu).
X = ramji .

?- grandmother(savita,X).
X = veni .

?- brother(X,Y).
X = amu,
Y = mahesh .

?- sister(X,Y).
X = veni,
Y = mitu .
```

SWI-Prolog (AMD64, Multi-threaded, version 8.2.3)
File Edit Settings Run Debug Help

```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
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).

?-
% d:/College Practical/sem-8 practical/AI/Prolog Testing/1_pr_b_family_tree.pl compiled 0.00 sec, 35 clauses
?- mother(X,veni).
X = jignasa.

?- father(amu,X).
X = veni .

?- father(amu,X),female(X).
X = veni .

?- grandfather(X,mitu).
X = ramji .

?- grandmother(savita,X).
X = veni .

?- brother(X,Y).
X = amu,
```

Fig 1.7 Family tree