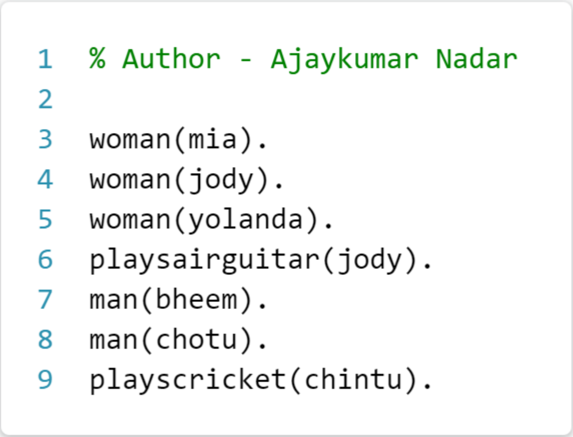
Knowledge Base:

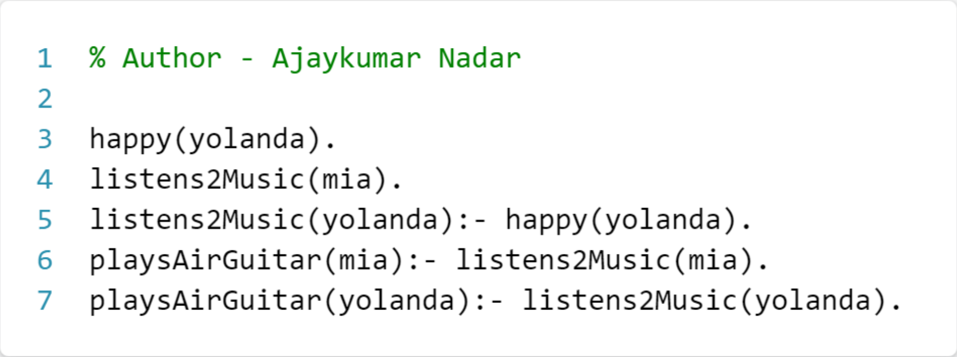


```prolog
% Author - Ajaykumar Nadar

woman(mia).
woman(jody).
woman(yolanda).
playsairguitar(jody).
man(bheem).
man(chotu).
playscricket(chintu).
```

Query & Output:

```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

1 ?- pwd.
% c:/users/ajay kumar/
true.

2 ?- cd('C:/Users/Ajay kumar/Desktop/SEIT-B/PCPF/Lab/Exp_6').
true.

3 ?- [prolog1].
true.

4 ?- woman(mia).
true.

5 ?- woman(jody).
true.

6 ?- woman(yolanda).
true.

7 ?- playsAirGuitar(jody).
Correct to: "playsairguitar(jody)"? yes
true.

8 ?- playsairguitar(mia)
.
false.

9 ?- playscricket(chintu).
true.

10 ?- man(Bheem).
Bheem = bheem.
```

Knowledge Base:

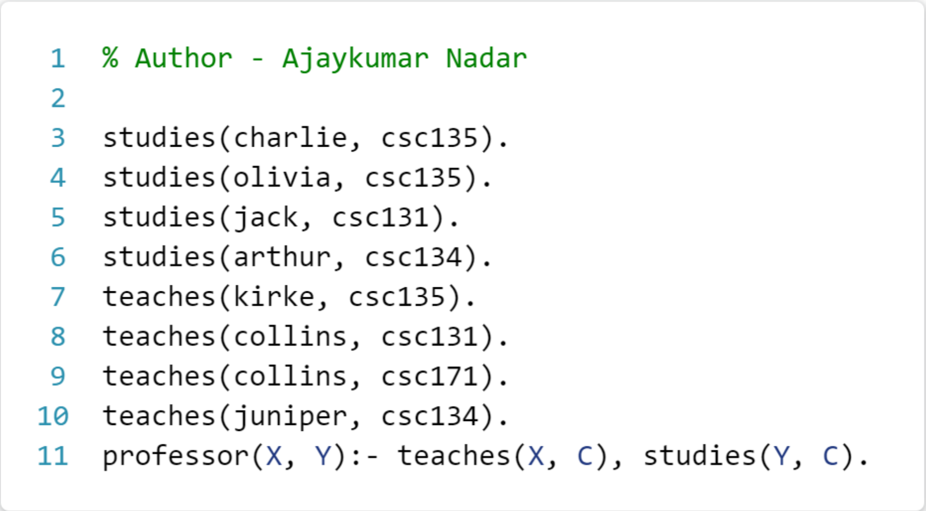


```
% Author - Ajaykumar Nadar

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

Query & Output:

```
1 ?- cd('C:/Users/Ajay kumar/Desktop/SEIT-B/PCPF/Lab/Exp_6').
true.

2 ?- [prolog2].
true.

3 ?- happy(mia).
false.

4 ?- listens2Music(mia).
true.

5 ?- listens2Music(yolanda).
true.

6 ?- happy(X).
X = yolanda.

7 ?- playsairguitar(Y).
Correct to: "playsAirGuitar(Y)"? yes
Y = mia .

8 ?- listens2Music(Z).
Z = mia .

9 ?-
```

Knowledge Base:



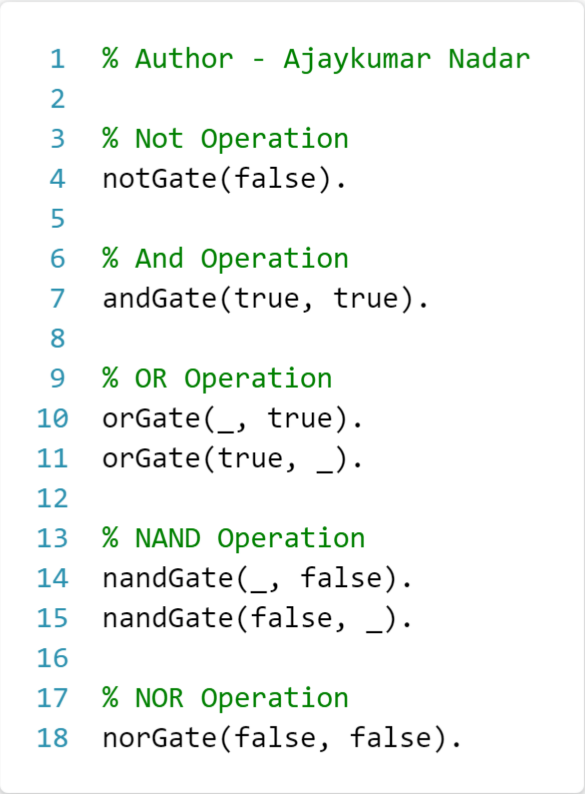
```
% Author - Ajaykumar Nadar

studies(charlie, csc135).
studies(olivia, csc135).
studies(jack, csc131).
studies(arthur, csc134).
teaches(kirke, csc135).
teaches(collins, csc131).
teaches(collins, csc171).
teaches(juniper, csc134).
professor(X, Y):- teaches(X, C), studies(Y, C).
```

Query & Output:

```
1 ?- cd('C:/Users/Ajay kumar/Desktop/SEIT-B/PCPF/Lab/Exp_6').
true.

2 ?- [prolog3].
true.

3 ?- studies(charlie, What).
What = csc135.

4 ?- professor(kirke, Students).
Students = charlie .

5 ?- teaches(Who, csc171).
Who = collins.
```

4. WAP in Prolog to implement the truth tables of the logical operations-NOT, AND, OR, NAND and NOR operations.

Knowledge Base:



```
% Author - Ajaykumar Nadar

% Not Operation
notGate(false).

% And Operation
andGate(true, true).

% OR Operation
orGate(_, true).
orGate(true, _).

% NAND Operation
nandGate(_, false).
nandGate(false, _).

% NOR Operation
norGate(false, false).
```

Query & Output:

```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
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).

1 ?- cd('C:/Users/Ajay kumar/Desktop/SEIT-B/PCPF/Lab/Exp_6').
true.

2 ?- [prolog4].
true.

3 ?- notGate(true).
false.

4 ?- and(false, true).
ERROR: Unknown procedure: and/2 (DWIM could not correct goal)

5 ?- andGate(false, true).
false.
```

```
6 ?- andGate(true, true).
true.

7 ?- orGate(false, false).
false.

8 ?- orGate(true, false).
true.

9 ?- nandGate(true, true).
false.

10 ?- nandGate(false, false).
true .

11 ?- norGate(true, true).
false.

12 ?- norGate(false, false).
true.
```

5 Implement the following logical operations

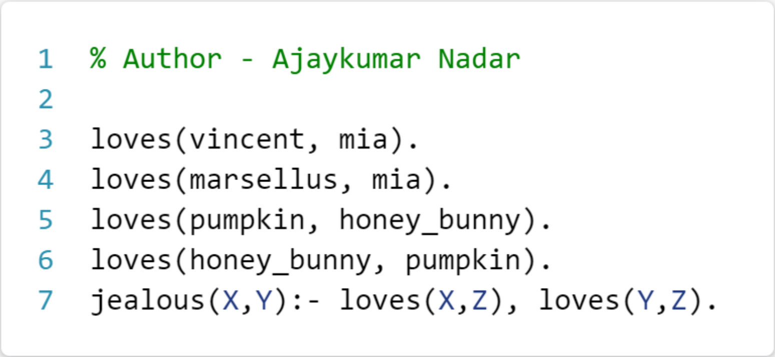
Query & Output:

```
1 ?- 6*6=:=36.
true.

2 ?- 10=8+3.
false.

3 ?- 10=8+2.
false.

4 ?- sqrt(36)+4=:=5*11-45.
true.

5 ?- 10=\=8+3.
true.

6 ?- 11=\=8+3.
false.

7 ?- 6+4==3+7.
false.

8 ?- 6<3;7 is 5+2.
true.

9 ?- not(111=\=8+3).
false.

10 ?- 111=\=8+3,11=\=3.
true.

11 ?- 11=\=8+3;11=\=3.
true.

12 ?- sqrt(36)+4=:=5*11-45.
true.

13 ?- sqrt(36)+4=:=5*11-45;false.
true .

14 ?- sqrt(36)+4=:=5*11-45,false.
false.

15 ?- not(6<3;7 is 5+2).
```

```
false.

16 ?- not(not(6<3;7 is 5+2)).
true.

17 ?- not(not(6<3;7 is 5+2)),true.
true.

18 ?- not(not(6<3;7 is 5+2)),false.
false.

19 ?- not(not(6<3;7 is 5+2));false.
true .

20 ?- not(not(6<3;7 is 5+2));true.
true .
```

4. WAP to create the following knowledge base loves(vincent,mia). loves(marsellus,mia). loves(pumpkin,honey_bunny). loves(honey_bunny,pumpkin). jealous(X,Y):- loves(X,Z), loves(Y,Z). Also generate five different queries to get results

Knowledge Base:



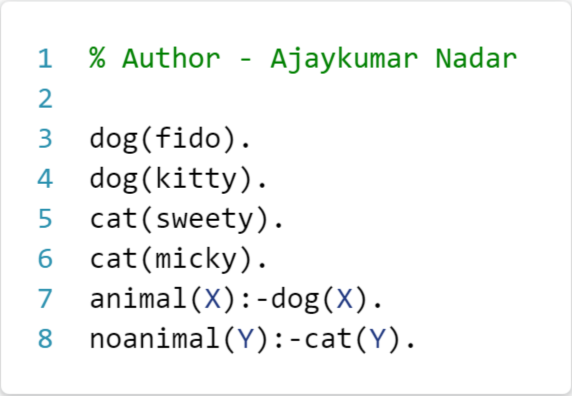
```
% Author - Ajaykumar Nadar

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

Query & Output:

```
1 ?- [prolog5].
true.

2 ?- loves(vincent, X).
X = mia.

3 ?- loves(X, mia).
X = vincent ;
X = marsellus.

4 ?- jealous(X, vincent).
X = vincent ;
X = marsellus ;
false.

5 ?- jealous(X, honey_bunny).
X = honey_bunny.

6 ?- jealous(X, Y).
X = Y, Y = vincent ;
X = vincent,
Y = marsellus ;
X = marsellus,
Y = vincent ;
X = Y, Y = marsellus ;
X = Y, Y = pumpkin ;
X = Y, Y = honey_bunny.
```

5. Write a program in Prolog to include the following facts and rules in the knowledge base.

Knowledge Base:



```prolog
% Author - Ajaykumar Nadar

dog(fido).
dog(kitty).
cat(sweety).
cat(micky).
animal(X):-dog(X).
noanimal(Y):-cat(Y).
```

Query & Output:

```
1 ?- [prolog6].
true.

2 ?- not(dog(fido)).
false.

3 ?- not(dog(kitty)).
false.

4 ?- not(cat(sweety)).
false.

5 ?- not(cat(micky)).
false.

6 ?- not(dog(fiddo)).
true.

7 ?- not(cat(sweety)).
false.

8 ?- not(cat(sweety)),not(cat(sweety)).
false.

9 ?- not(cat(sweety));not(cat(miccy)).
true.

10 ?- not(cat(sweety));not(cat(micky)).
false.
```