EXP:NO:10 NAME:MOHAMMED FAISAL S.M

REG:NO:231801105

DEPT:AIDS-B

INTRODUCTION TO PROLOG

SOURCE CODE:

KB1:

woman(mia).

woman(jody).

woman(yolanda).

playsAirGuitar(jody).

party.

Query 1: ?-woman(mia).

Query 2: ?-playsAirGuitar(mia).

Query 3: ?-party.

Query 4: ?-concert.

OUTPUT: -



KB2:

happy(yolanda).

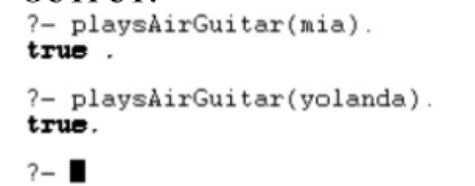
listens2music(mia).

Listens2music(yolanda):-happy(yolanda).

playsAirGuitar(mia):-listens2music(mia).

playsAirGuitar(Yolanda):-listens2music(yolanda).

OUTPUT: -



KB3:

likes(dan,sally).

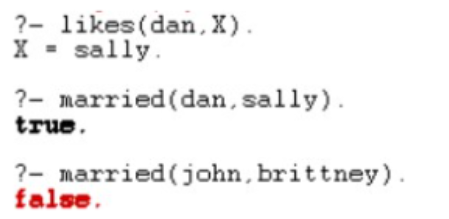
likes(sally,dan).

likes(john,brittney).

married(X,Y) :- likes(X,Y) , likes(Y,X).

friends(X,Y) :- likes(X,Y) ; likes(Y,X).

OUTPUT: -



KB4:

food(burger).

food(sandwich).

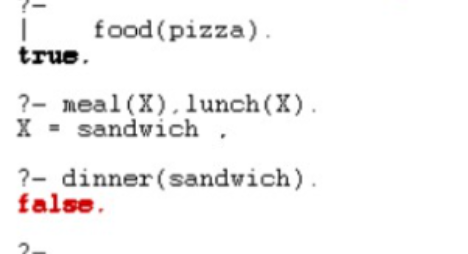
food(pizza).

lunch(sandwich).

dinner(pizza).

meal(X):-food(X).

OUTPUT:



KB5:

owns(jack,car(bmw)).

owns(john,car(chevy)).

owns(olivia,car(civic)).

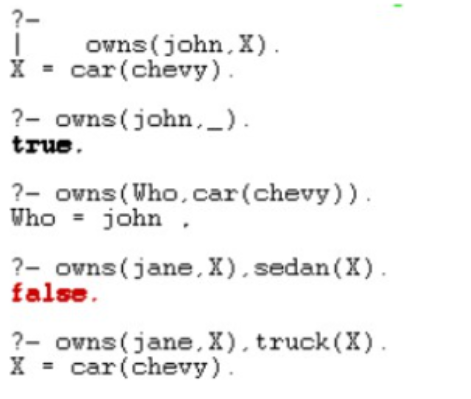
owns(jane,car(chevy)).

sedan(car(bmw)).

sedan(car(civic)).

truck(car(chevy)).

OUTPUT:



KB6: Find minimum maximum of two numbers

find\_max(X,Y,X):-X&gt;=Y,!.

find\_max(X,Y,Y):-X&lt;Y.

find\_min(X,Y,X):-X=&lt;Y,!.

find\_min(X,Y,Y):-X&gt;Y.

Output:

| ?- find\_max(100,200,Max).

Max = 200

yes

| ?- find\_max(40,10,Max).

Max = 40

yes

| ?- find\_min(40,10,Min).

Min = 10

yes

| ?- find\_min(100,200,Min).

Min = 100

yes

| ?-

KB7:

Here are some simple clauses.

likes(mary,food).

likes(mary,wine).

likes(john,wine).

likes(john,mary).

The following queries yield the specified answers.

| ?- likes(mary,food).

yes.

PAGE NUMBER:

| ?- likes(john,wine).

yes.

| ?- likes(john,food).

no.

How do you add the following facts?

1. John likes anything that Mary likes

2. John likes anyone who likes wine

3. John likes anyone who likes themselves

% Existing facts

likes(mary, food).

likes(mary, wine).

likes(john, wine).

likes(john, mary).

% New facts

likes(john, X) :- likes(mary, X). % John likes anything that Mary likes

likes(john, Y) :- likes(Y, wine). % John likes anyone who likes wine

likes(john, Z) :- likes(Z, Z). % John likes anyone who likes themselves

% Queries and their answers

% Query: likes(mary, food).

% Answer: yes.

% Explanation: Mary likes food (existing fact).

% Query: likes(john, wine).

% Answer: yes.

% Explanation: John likes wine (existing fact).

% Query: likes(john, food).

% Answer: no.

% Explanation: John does not like food (not explicitly defined).

% Existing facts

likes(mary, food).

likes(mary, wine).

likes(john, wine).

likes(john, mary).

% New facts and rules

likes(john, X) :- likes(mary, X).

% John likes anything that Mary likes.

% This rule means that if Mary likes something X, then John also likes X.

likes(john, Y) :- likes(Y, wine).

% John likes anyone who likes wine.

% This rule means that if someone Y likes wine, then John also likes Y.

likes(john, Z) :- likes(Z, Z).

% John likes anyone who likes themselves.

% This rule means that if someone Z likes themselves, then John also likes Z.

% Queries and their answers

% Query: likes(mary, food).

% Answer: yes.

% Explanation: Mary likes food (existing fact).

% Query: likes(john, wine).

% Answer: yes.

% Explanation: John likes wine (existing fact).

% Query: likes(john, food).

% Answer: no.

% Explanation: John does not like food (not explicitly defined).