**TEAM - D**

**LAB -8 : KNOWLEDGE REPRESENTATION**

EXAMPLE 1

CODE:

FACTS

owns(jack, car(bmw)).

owns(john, car(benz)).

owns(tom, car(maruti)).

owns(jane, car(benz)).

sedan(car(bmw)).

sedan(car(maruti)).

sports(car(benz)).

QUERIES

?- owns(john, X). // What does john own?

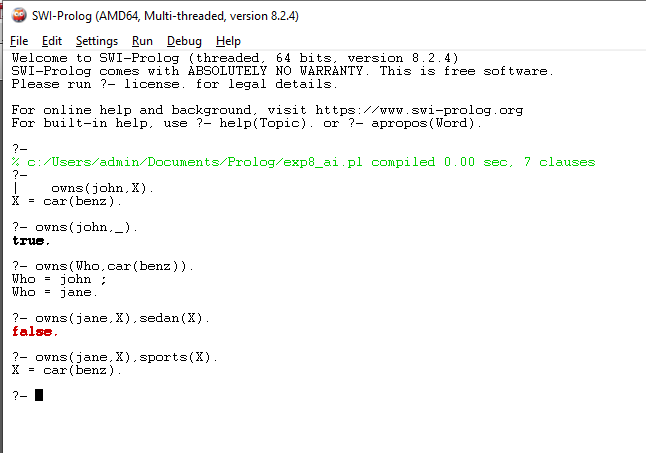
?- owns(john, \_ ). // Does john own something?

?- owns(Who, car(benz)). // Who owns car benz?

?- owns(jane, X), sedan(X). // Does jane own sedan?

?- owns(jane, X), sports(X). // Does jane own sports car?

OUTPUT:



EXAMPLE 2

FACTS

food(burger).

food(sandwich).

food(pizza).

lunch(sandwich).

dinner(pizza).

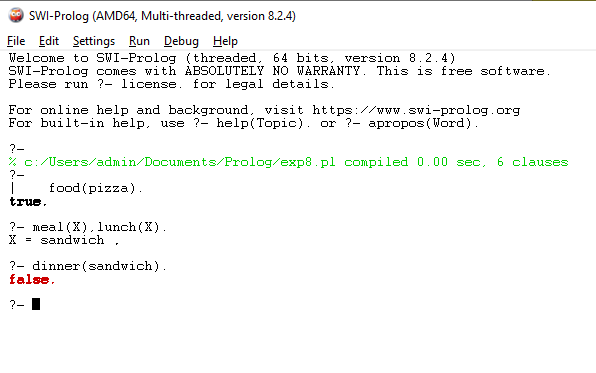
RULES

meal(X) :- food(X).

QUERIES

|  |  |
| --- | --- |
| ?- food(pizza). | // Is pizza a food? |
| ?- meal(X), lunch(X). | // Which food is meal and lunch? |
| ?- dinner(sandwich). | // Is sandwich a dinner? |

OUTPUT:



**Result:**

Thus, Knowledge Representation was successfully implemented and executed in SWI Prolog.