INTRODUCTION TO ARTIFICIAL INTELLIGENCE

LAB ASSIGNMENT 8

Use of PROLOG for Logic Programming
Innovator's Game

SUBMITTED BY:

Group No. 04

Sandhya Yadav - 201652023

Payal Mahisaniya - 201652015

Pooja Gurjar - 201652016

PROBLEM STATEMENT:

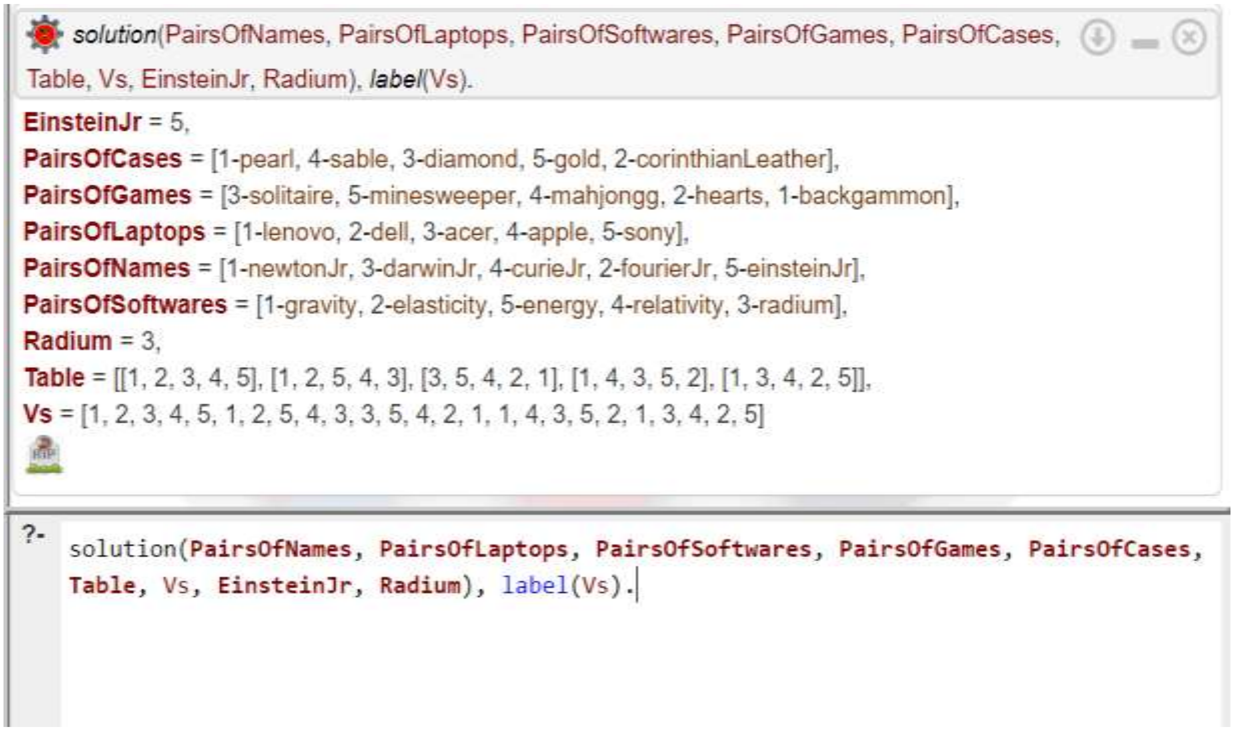
IIIT Vadodara is a home to terrific innovators. Recently, Einstein Jr and four other members of the innovators club purchased custom-made laptops, each of which is a different brand, and brought them to the institute for a bit of show and brag. Each computer's case is inlaid with a different expensive material and came installed with a different software (one of them being Radium) to help innovators develop business plans. After displaying their fancy laptops, each of the five members sheepishly admitted that he or she mainly uses the laptop to play games. However the five innovators expect to get more use from their laptop upon conquering the next frontier-tech support! From the information provided, determine the innovator who owns each brand of laptop, the material inlaid in each computer's case, the software that **came** installed with each laptop, and the game each innovator plays on his or her computer.

1. Mr. Newton Jr owns the laptop with a pearl inlaid case (which came installed with Gravity software). The *Lenovo* laptop is used mainly to play **backgammon**.
2. The owner of the *Dell* laptop (which came installed with Elasticity program) uses it mainly to play **hearts**.
3. Mr. Darwin Jr's *Accer* laptop isn't the one that came installed with the Energy program (which isn't the one installed on the *Apple* laptop).
4. The owner of the computer that came installed with the Relativity program isn't the one who plays a lot of **solitaire**.
5. Ms. Curie Jr uses her laptop (which sports sable inlays) mainly to play **mah-jongg**. Mr. Fourier Jr laptop is inlaid with rich Corinthian leather.
6. The owner of the laptop with the diamond inlaid case isn't the one who plays a lot of **minesweeper**. The *Sony* laptop has a gold-inlaid case.

Code:

```
:-use_module(library(clpfd)).
:-use_module(library(pairs)).

solution(Pair1, Pair2, Pair3, Pair4, Pair5, Table, Vs, EinsteinJr, Radium)
:-
        Table   = [Laptops, Softwares, Games, Cases, Names],
        Laptops  = [Lenovo,Dell,Acer,Apple,Sony],
      Names = [NewtonJr, DarwinJr, CurieJr, FourierJr, EinsteinJr],
      Softwares = [Gravity, Elasticity, Energy, Relativity, Radium],
      Games = [Solitaire, Minesweeper, Mahjongg, Hearts, Backgammon],
      Cases = [Pearl, Sable, Diamond, Gold, CorinthianLeather],

      Laptop  = [lenovo,dell,acer,apple,sony],
      Name = [newtonJr, darwinJr, curieJr, fourierJr, einsteinJr],
      Software = [gravity, elasticity, energy, relativity, radium],
      Game = [solitaire, minesweeper, mahjongg, hearts, backgammon],
      Case = [pearl, sable, diamond, gold, corinthianLeather],

      pairs_keys_values(Pair1, Names,Name),
      pairs_keys_values(Pair2, Laptops, Laptop),
      pairs_keys_values(Pair3, Softwares, Software),
      pairs_keys_values(Pair4, Games, Game),
      pairs_keys_values(Pair5, Cases, Case),
        maplist(all_distinct, Table),
        append(Table, Vs),
      Vs ins 1..5,

      NewtonJr #= Pearl,
      NewtonJr #= Gravity,
      Lenovo #= Backgammon,
      Dell #= Hearts,
      Dell #= Elasticity,
      DarwinJr #= Acer,
      Acer #\= Energy,
      Apple #\= Energy,
      Relativity #\= Solitaire,
      CurieJr #= Sable,
      CurieJr #= Mahjongg,
      FourierJr #= CorinthianLeather,
      Diamond #\= Minesweeper,
      Sony #= Gold.
```

Output:



```
solution(PairsOfNames, PairsOfLaptops, PairsOfSoftwares, PairsOfGames, PairsOfCases,
Table, Vs, EinsteinJr, Radium), label(Vs).
EinsteinJr = 5,
PairsOfCases = [1-pearl, 4-sable, 3-diamond, 5-gold, 2-corinthianLeather],
PairsOfGames = [3-solitaire, 5-minesweeper, 4-mahjongg, 2-hearts, 1-backgammon],
PairsOfLaptops = [1-lenovo, 2-dell, 3-acer, 4-apple, 5-sony],
PairsOfNames = [1-newtonJr, 3-darwinJr, 4-curieJr, 2-fourierJr, 5-einsteinJr],
PairsOfSoftwares = [1-gravity, 2-elasticity, 5-energy, 4-relativity, 3-radium],
Radium = 3,
Table = [[1, 2, 3, 4, 5], [1, 2, 5, 4, 3], [3, 5, 4, 2, 1], [1, 4, 3, 5, 2], [1, 3, 4, 2, 5]],
Vs = [1, 2, 3, 4, 5, 1, 2, 5, 4, 3, 3, 5, 4, 2, 1, 1, 4, 3, 5, 2, 1, 3, 4, 2, 5]
```

```
?- solution(PairsOfNames, PairsOfLaptops, PairsOfSoftwares, PairsOfGames, PairsOfCases,
   Table, Vs, EinsteinJr, Radium), label(Vs).
```

Interpretation of the Result: From the above output we interpret that the solution function returns 5 pairs, a table, Vs (labels), and the corresponding label of EinsteinJr and Radium because they are not given in the rules. The 5 pairs are mapping of labels and their corresponding entities. For Example we can say **PairsOfName** clarifies mapping of each name and its label. The final result is a list of tuples in which each tuple has the same label. For example, all the entities with label 1 become a set i.e. (NewtonJr, Lenovo, Gravity, Pearl, Backgammon).

Approach: We use **clpfd** library which is an acronym for Constraint Logic Programming over Finite Domains. We write all the given constraints and derived the final relationship between the given entities. The library used inference methods like forward chaining and unification to derive all the possible relationship under given constraint.