Practical 3

1. Imagine that the following knowledge base describes a maze. The facts determine which points are connected, i.e., from which point you can get to which other point in one step. Furthermore, imagine that all paths are one-way streets, so that you can only walk them in one direction. So, you can get from point 1 to point 2, but not the other way round.

```
connected(1,2).

connected(3,4).

connected(5,6).

connected(7,8).

connected(9,10).

connected(12,13).

connected(13,14).

connected(15,16).

connected(17,18).

connected(19,20).

connected(4,1).

connected(6,3).

connected(4,7).

connected(6,11).

connected(14,9).

connected(11,15).

connected(16,12).

connected(14,17).

connected(16,19).
```

Write a predicate `path/2` that tells you from which point in the maze you can get to which other point when chaining together connections given in the above knowledge base.

```
path(X,Y) :- connected(X,Y).

path(X,Y) :- connected(X,Z), path(Z,Y).
```

Can you get from point 5 to point 10?

```
?- path(5,10).
Yes
```

Which other point can you get to when starting in point 1?

```
?- connected(1,X).
X = 2 ;
No
```

And which points can be reached from point 13?

```
?- connected(13,X).
X = 14 ;
No
```

2. We are given the following knowledge base of travel information:

```
byCar(auckland,hamilton).
byCar(hamilton,raglan).
byCar(valmont,saarbruecken).
byCar(valmont,metz).

byTrain(metz,frankfurt).
byTrain(saarbruecken,frankfurt).
byTrain(metz,paris).
byTrain(saarbruecken,paris).


byPlane(frankfurt,bangkok).
byPlane(frankfurt,singapore).
byPlane(paris,losAngeles).
byPlane(bangkok,auckland).
byPlane(losAngeles,auckland).
```

Write a predicate `travel/2` which determines whether it is possible to travel from one place to another by 'chaining together' car, train, and plane journeys. For example, your program should answer 'yes' to the query `travel(valmont,raglan).`

```
travel(X,Y) :- byCar(X,Y).

travel(X,Y) :- byTrain(X,Y).

travel(X,Y) :- byPlane(X,Y).

travel(X,Y) :- travel(X,Z), travel(Z,Y).
```

3. So, by using `travel/2` to query the above database, you can find out that it is possible to go from Vamont to Raglan. In case you are planning a travel, that's already very good information, but what you would then really want to know is how exactly to get from Valmont to Raglan. Write a predicate `travel/3` which tells you how to travel from one place to another. The program should, e.g., answer 'yes' to the query `travel(valmont,paris,go(valmont,metz,go(metz,paris)))` and `X = go(valmont,metz,go(metz,paris,go(paris,losAngeles)))` to the query `travel(valmont,losAngeles,X).`

```
travel(X,Y,go(X,Y)) :- byCar(X,Y).

travel(X,Y,go(X,Y)) :- byTrain(X,Y).

travel(X,Y,go(X,Y)) :- byPlane(X,Y).

travel(X,Y,go(X,Z,G)) :- travel(X,Z,go(X,Z)), travel(Z,Y,G).
```

4. Extend the predicate `travel/3` so that it not only tells you via which other cities you have to go to get from one place to another, but also *how*, i.e. by car, train, or plane, you get from one city to the next.

```
travel(X,Y,go(X,Y,car)) :- byCar(X,Y).

travel(X,Y,go(X,Y,train)) :- byTrain(X,Y).

travel(X,Y,go(X,Y,plane)) :- byPlane(X,Y).

travel(X,Y,go(X,Z,V,G)) :- travel(X,Z,go(X,Z,V)), travel(Z,Y,G).
```