Prolog Lab Test

You have 1 hour and 45 minutes to attempt this test. When finished, put all you code into a single Prolog text file named **labtest.pl** and submit it to Webcourses. Do not submit more than 1 Prolog file.

1. Represent the following simplified knowledge of eligible footballers in Prolog in a way you deem appropriate.
   - Any person is eligible for the local football team if they are aged between 18 and 32 inclusive and their fitness level is excellent.
   - Someone whose fitness level is good and who drives a car and who drink alcohol either not at all or moderately is also eligible.
   - Finally someone is eligible if their father is important, no matter what their fitness level.

   Age restrictions apply to all candidates. Use predicates like **person(name, age, gender, fitness_level)**, **eligible(Name)**, **drives(name, carType)**, **isImportant(name)**, **drinks(name, level)**. Populate you model with some facts to test it.

   (25 marks)

2. Given the following nonsensical facts about 1-way cycling and moped (small motorbike) lanes:

```
bicycleLane(dublin, kilkenny, 120).
bicycleLane(dublin, carlow, 80).
bicycleLane(kilkenny, waterford, 85).
bicycleLane(mallow, carlow, 110).
bicycleLane(cork, mallow, 38).
bicycleLane(waterford, youghal, 110).
bicycleLane(youghal, cork, 30).
bicycleLane(dublin, athlone, 105).
bicycleLane(athlone, limerick, 135).
bicycleLane(limerick, cork, 75).
moped(dublin, galway, 200).
moped(galway, limerick, 110).
moped(limerick, ennis, 30).
moped(limerick, cork, 95).
```

   Write a predicate **canGoFromTo(A, B)** which will tell you if you can ride a bicycle or moped from town A to town B even if you have to pass thru other towns. How many ways can you get from Dublin to Cork? Is it possible to get to Limerick from Carlow? Put your answers in as comments in your code after running it.

   (15 marks)

3. If it is possible to cycle from A to B, you would like to know the total distance. To do this add a new rule **canGoFromTo(A, B, Distance)** which is an adaptation of canGoFromTo/2 so that the overall length of a possible journey from A to B can be recorded. You should have two different canGoFromTo() rules now with arities 2 and 3.

   (10 marks)

4. Modify canGoFromTo/3 from Q4 so that is also records the route taken, i.e. **canGoFromTo(A, B, Distance, Route)**

   (15 marks)

5. Write list predicates to:

   addTo(X, L, Lnew) where addTo(3, [1,3,0], Lnew) gives Lnew = [4,6,3]

   removeAll(X, L, Lnew) where removeAll(b, [12, b, sat, 32, a, b, 1], Lnew) gives Lnew = [12, sat, 32, a,1]

   freq(X, L, N) where freq(a, [1, a, 4, ba, a], N) gives N = 2

   listMax(L, X) where listMax([2,5,4,0], X) gives X = 5

makePairs(L, L2) where makePairs([1.3, 2, -1, 0.7], L2) gives L2 = [(1.3, 2.6), (2,4), (-1,-2), (0.7,1.4)] where the 2nd value of each pair is double the 1st value.

mergeLists(L1, L2, L) which merges two already sorted lists L1 and L2 into another sorted list L. For example mergeLists([3,5,6,10], [2,7], L) gives L = [2,3,5,6,7,10]. (10 marks)

(35 marks)