Exercise 1 (8 points)

Horses run faster than rabbits. Dogs run faster than rabbits. Because of its name, we know that Fury is either a horse or a dog. Bunny is a rabbit. Arrow is a greyhound.

- (a) Represent the above formulae in first order logic.
- (b) Transform them in CNF and prove by resolution that Fury is faster than Bunny. Or else show that it is not logically entailed.
- (c) Is Arrow faster than Bunny? If it is, prove it. Otherwise, specify whether you can add some knowledge in order to prove it (except for the trivial addition of $faster(Arrow, Bunny)$).
- (d) Represent in first order logic the sentences:
  "Italian soccer players are better than french ones" (don't argue if you do not believe it)
  "The best Italian soccer player is better than some French soccer player"

Exercise 2 (4 points)

Describe the main cases of the unification algorithm. Tell whether the following two expressions unify, showing step-by-step the execution of the unification algorithm (constants and functions are lower case, variables are capitalized):
$cons(a, cons(b, cons(X, [])))$ and $cons(Y, cons(Y, cons(b, [])))$

Exercise 3 (4 points)

An associative list is a list all of whose elements are 2-element lists; the first element of each element of the associative list is an atom, the second one is an arbitrary term. Example AL=[[a,f(0)],[b,f(1)],[c,f(2)]] is an associative list. Use `atom(X)` to test whether the argument is an atom.

- (a) Write a PROLOG program that checks whether its argument is an associative list.
- (b) Write a PROLOG program `FINDT` that, given in input an associative list AL, and atom A returns a term T, corresponding to the second element of the element of AL whose first element is A. Assume that every atom in the AL appears only once. Example: the result of `FINDT([[a,f(0)],[b,f(1)],[c,f(2)]],b,T)` is T=f(1).
- (c) [**EXTRA QUESTION tbd only after having answered everything else**]
  Write a PROLOG program `SUBS` that, given in input an associative list AL, and a list of atoms L, returns the result of replacing the occurrences of the atoms that are in AL with the terms associated to them. Assume that every atom in AL appears only once. Example: the result of `SUBS(AL,[a,b,a,d],L)` is L=[f(0),f(1),f(0),d].