TP 6

Exercise 1: Write a predicate last(List,X) which is true only when List is a list that contains at least one element and X is the last element of that list. Do this in two different ways:

1. Define last/2 using the predicate rev/2 discussed in the text.
2. Define last/2 using recursion.

Exercise 2 Write a predicate swapfl(List1,List2) which checks whether List1 is identical to List2 , except that the first and last elements are exchanged. Here's where append/3 could come in useful again, but it is also possible to write a recursive definition without appealing to append/3 (or any other) predicates.

Exercise 3 Here is an exercise for those of you who like logic puzzles.

There is a street with three neighbouring houses that all have a different colour, namely red, blue, and green. People of different nationalities live in the different houses and they all have a different pet. Here are some more facts about them:

- The Englishman lives in the red house.
- The jaguar is the pet of the Spanish family.
- The Japanese lives to the right of the snail keeper.
- The snail keeper lives to the left of the blue house.

Who keeps the zebra? Don't work it out for yourself: define a predicate zebra/1 that tells you the nationality of the owner of the zebra!

(Hint: Think of a representation for the houses and the street. Code the four constraints in Prolog. You may find member/2 and sublist/2 useful.)