Programming in Prolog - List of Exercises

1. When manipulating lists there are several basic predicates that every Prolog programmer would like to have as built-in predicate. Write the programs for "find the last element of a list", "check consecutive elements of a list", "erase one element from a list and generate another list", and "erase all ocurrences of an element from a list and generate a new list".

2. This problem is a sophisticated application of list concatenation. It pretends to build, on the basis of a list of French words (for instance, animal names), a list of new words, each one obtained from the ancestors by juxtaposition of the words whose final characters are the first ones of another. For example the words VACHE (cow) and CHEVAL (horse) give the word VACHEVAL (mutation problem). Write a program that produce the "mutations" for the following database of animals: alligator, tortue, caribou, ours, cheval, vache, lapin (French words for crocodile, turtle, caribou, bear, horse, cow and rabbit).

3. Write a program to simplify ("flatten") a list whose members are lists, and to transform it into a list of single members, containing no lists as members.

4. Define a set of predicates for manipulating queues. Consider the following queue:

   head $\rightarrow$ [a] $\rightarrow$ [b] $\rightarrow$ [c] $\leftarrow$ rear

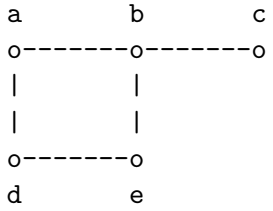
   Write the goal statements for deleting the head element of a queue, for inserting an element at the rear of a queue, and for inserting a new element into the queue.

5. Write a simple Propositional Calculus Theorem Prover, covering equivalence, implication, disjunction, conjunction and negation. (Suggestion: define operators like: op(700,xfy,<=>), op(650,xfy,=>),...).

6. Define the roman to arabic number conversion.

7. Write a program to solve the N-queens problem. The goal consists in finding all the ways of placing N queens on a NxN chess board so that no queen attacks another, where two queens are said to attack each other if they are positioned along a common row, column or diagonal.

8. A road map is represented by the following graph:



   Write a program to find the shortest path between two towns based upon the best-first strategy. Write another program based upon the breadth-first strategy.

9. Write a program to tell whether a sentence is semantically correct or not. Expand the program so that it can accept correct sentences (learn) for which it has previously given a wrong answer. The final program provides the user with a tool that allows to both ask questions and enter information in the same natural declarative way, i.e. questions and assertions.

10. Write a program for designing an architectural unit obeying the following specifications:

    - two rectangular rooms.
    - Each room has a window and interior door.
    - Rooms are connected by interior door.
    - One room also has an exterior door.
    - A wall can have only one door or window.
    - No window can face north.
    - Windows cannot be on opposite sides of the unit.

11. Write a simple grammar able to parse sentences, such as "John ate the cake" and to construct their corresponding deep tree structure (Suggestion: use the definite clause grammar (DCG) formalism).

12. A program is required to examine a piece of text and produce a list, in alphabetical order, of all the distinct words which appear in the text, e.g. the examination of the input `the black dog chased the black cat` would produce the corresponding output `black cat chased dog the`.