Solutions

Assignment 3

Exercise 1

Write a PROLOG program counting the elements of a list of lists.

```
countListElems([], 0).

countListElems([X|Xs], N):-
        countListElems(X, N1),
        countListElems(Xs, N2),
        N is N1+N2.

countListElems(_X, 1).
```

Exercise 2

Write a PROLOG program which implements member for a binary tree.

```
hasElement(X, tree(X, _Left, _Right)).
hasElement(X, tree(_Element,Left,_Right)):-
        hasElement(X, Left).
hasElement(X, tree(_Element, _Left, Right)) :-
        hasElement(X, Right).
```

Exercise 3

Write a PROLOG program which returns a list containing all the nodes at a given depth D of a binary tree.

```
% printDeepElems(binaryTree,nodeList,initDepth,depth)
printDeepElems(void,[],_Cr,_C). %base step 1
printDeepElems(tree(_X,_LT,_RT),[_X],_C,_C). %base step
printDeepElems(tree(_X,LT,RT),L,Cr,C):- % inductive ste
        Cr<C, Cr1 is Cr+1,
        printDeepElems(LT,LL,Cr1,C),
        printDeepElems(RT,RL,Cr1,C),
        append(LL,RL,L).
```

Exercise on binary trees

a) Consider the PROLOG terms representing the binary trees whose nodes are labelled by a constant symbol and, in addition, store the depth of the node. Write a PROLOG program that returns true if its argument is a binary tree as above specified.

```
binary_tree(void).
binary_tree(tree(node(_Element,_Prof),Left,Right)) :-
        binary_tree(Left), binary_tree(Right).
```

Binary trees (b)

Write a PROLOG program that, given in input a binary tree and a constant, returns the depth of a node containing the given constant.

```
trovaprof(tree(node(X,P),_L,_R),X,P).
trovaprof(tree(node(_XX,_PP),Left,Right),X,P):-
         trovaprof(Left,X,P).
trovaprof(tree(node(_XX,_PP),Left,Right),X,P):-
         trovaprof(Right,X,P).
```

(home) check whether the program admits more than a solution and in such a case add cuts so that only one solution is returned.

Binary trees (c)

Write a PROLOG program that, given in input a binary tree without the depth information on the nodes and a constant, returns the depth of a node containing the given constant.

```
prof(tree(node(X,_Y),L,R),X,P,P).
prof(tree(node(_Element,_Prof),Left,Right),X,Y,R) :-
     Z is Y+1,
     prof(Left,X,Z,R).
prof(tree(node(_Element,_Prof),Left,Right),X,Y,R) :-
     Z is Y+1,
     prof(Right,X,Z,R).
```

Binary trees (d)

Write a PROLOG program that, given in input a binary tree without the depth information on the nodes, returns an isomorphic binary tree with the depth information stored in the nodes.

```
assegnaprof(void,P,void).
assegnaprof(tree(node(_X,_Y),L,R),P,
            tree(node(_X,P),L1,R1)) :-
       Z is P+1,
       assegnaprof(L,Z,L1),
       assegnaprof(R,Z,R1).
```