Homework 2

a) Consider predicate *trees(.)* which is satisfied for a finite set of binary trees (using the encoding from the first lecture). Define predicate *trees2nodes(., ., .)* which converts the binary trees from *trees(.)* into 3-tuples of labels which correspond to the nodes of the trees. For example, for
trees(.) = {[n(a, n(b, 0, n(d, 0, 0)), n(c, n(e, 0, 0), 0))]}
the predicate trees2nodes(., ., .) holds for
{[[a, b, c], [b, 0, d], [c, e, 0], [d, 0, 0], [e, 0, 0]]}.

b) Consider predicate *nodes(., ., .)*, which is satisfied for nodes of a binary tree (or binary trees). Implement predicate *node2tree(.)*, which does a conversion opposite to the task a), i.e. it converts 3-tuples of labels from *node(., ., .)* into a set of (maximal) binary trees. For examples, for
nodes(., ., .) = {[[a, b, c], [b, 0, d], [c, e, 0], [d, 0, 0], [e, 0, 0]]}
the predicate *nodes2trees(.)* holds for
{[n(a, n(b, 0, n(d, 0, 0)), n(c, n(e, 0, 0), 0))]}.

Implement the tasks a) and b) in SWI-Prolog (http://www.swi-prolog.org), hand out your solution (hw2.pl) with comments in (hw2.pdf) via email to me.