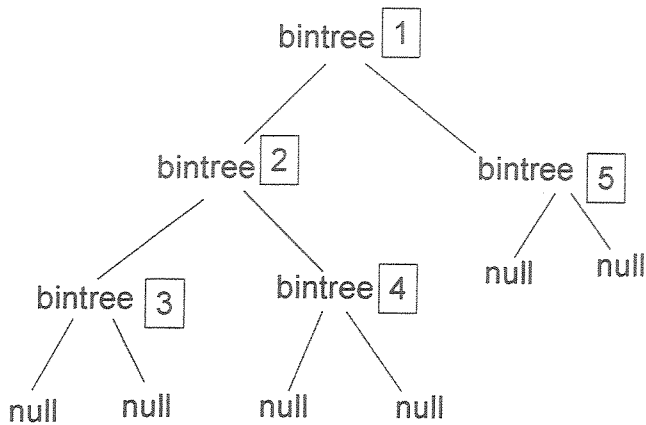


c) (20 points)\*\* In your Prolog project you had to write a predicate that evaluated a parsed expression stored in an abstract syntax tree by walking over the tree. Assume we are building binary trees in Prolog like the one shown below, using the following conventions:

--Each internal node has an integer label (shown in parentheses below) and less than or equal to 2 child nodes

--Each internal node is referred to by the functor **bintree**

--Each leaf node is represented by a Prolog literal **null**.



`bintree(1, bintree(2, bintree(3,null,null), bintree(4,null,null)), bintree(5,null,null) )`

Write the Prolog clauses for the **walk** predicate which performs a preorder traversal of such binary trees and returns a list of the node labels encountered in preorder.

`walk ( bintree (V,L,R), [V|Rest] ) :-`  
`walk (L,LL), walk (R,RL),`  
`append (LL,RL, Rest).`  
`walk (null, []).`

OR

`walk (bintree (V,L,R), [V|Other], End) :-`  
`walk (L, Other, ROther),`  
`walk (R, ROther, End).`  
`walk (null, A, A).`