The deadline for finishing Round 2 of TRAKLA2 problems is **Sunday 8 February**!

1. Program Java methods to compute the
   
   (a) height
   (b) number of leaves

   of a binary tree. *Hint:* using recursion will help.

2. Give an algorithm (in pseudocode) that performs an inorder tree walk without using recursion.

3. In correspondence to a binary tree, let’s form a *k-tree*, in which each node has at most *k* children. How many nodes in total can there be in a *k*-tree with height *h*? If you place *n* nodes in a *k*-tree, so that it becomes as low as possible, how high will it become?

4. Write the following unambiguous binary tree (it is not a search tree):
   
   Its nodes listed by preorder tree walk are: 10, 15, 9, 13, 6, 12, 3, 7, 16, 19, 5, 20
   Its nodes listed by inorder tree walk are: 9, 15, 6, 13, 12, 10, 7, 3, 19, 5, 16, 20