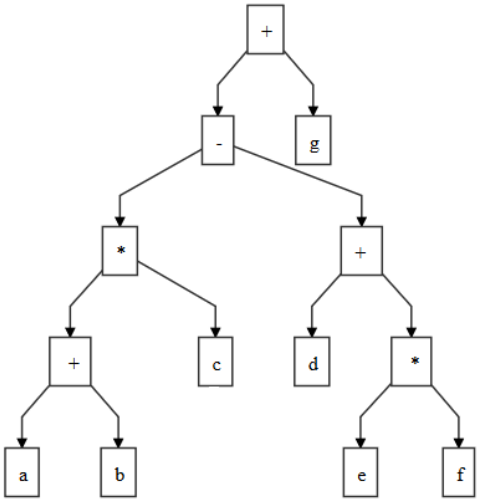
1. Build the binary tree for an arithmetic expression that contains the operators +, -, *, /. **Given**: the postfix notation of the expression.

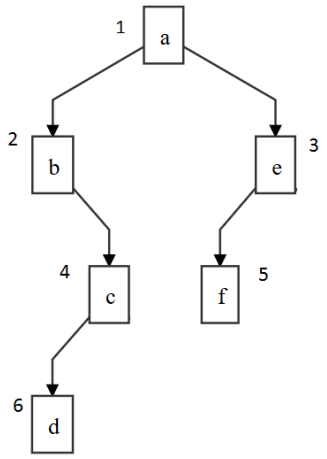
   e.g.: (a + b)* c – (d + e * f) + g =>
   Postfix notation: ab+c*def*+-g+

   The corresponding binary tree is:



   If we traverse the tree in postorder, we will get the postfix notation.

2. Generate the table with information from a binary tree. Node numbering is done according to levels.
   - The problem requires two things
     - Assign a number to every node (considering the levels)
     - Fill in the table with the information about the node, considering the assigned numbers



	1	2	3
	Info	Index Left	Index Right
1	a	2	3
2	b	-1	4
3	e	5	-1
4	c	6	-1
5	f	-1	-1
6	d	-1	-1

- Divide the solution into 2 functions: *addNumbers* (implemented non-recursively) and *buildTable* (implemented recursively)
- Assume that each Node has a field *nr:Integer* (we are going to store the number of a node here).

3. Given a binary tree that represents the ancestors of a person up to the n^{th} generation, where the left subtree represents the maternal line and the right subtree represents the paternal line:
   a) Display all the females from the tree (root can be either male of female)
      a. a, x, z, t (assuming root is female)
   b) Display all ancestors of degree *k* (root has degree 0)
      a. K = 2 : z, w, t, s