Lab 7: AVL Tree

7.1. For an AVL tree to store integer values, implement the following requirement:

- Create data structure for AVL Tree
- Initialize empty tree
- Insert some value to tree
- Print tree to console screen in pre-order, in-order, post-order
- Delete some value from tree
- Calculate tree height.
- Identify whether a given value is exist in tree

Students must present some functions such as single rotate left, single rotate right, double rotate left, double rotate right to balance tree.

7.2 Write a program to determine if given binary trees are AVL Tree. Suppose that trees only store integer number type.

Your program read trees in input.txt and output yes/no AVL Tree. In input.txt, first row is number of tree. Next rows are trees with each tree will be on a line and in pre-order. Each integer value is separated by a space.

For example, *input.txt*:

3

1 2 3 4 5

5 4 3 2 1

4 2 1 3 5

Ouput.txt:

no

no

yes

7.3. Write a program to test whether all leaves of a AVL tree have the same depth

7.4. Write a program to find the least common ancestor for any two given nodes in AVL.