ASSIGNMENT 2

Write MyTree class in Java using Binary Search Tree data structure to manage information about persons. Variables used to store information about a person are:

- name - the name of a person (character String) , which is **the key of the tree**.
- age - the age of a person (integer value).

You should write the MyTree class with the following methods:

1. **void insert(String xName, int xAge)**: check if xName contains 'B' or age > 10 then do nothing, otherwise, insert that person information to the tree.
   For example, the input for insert method can be:
   (A6,1) (A2,5) (B6,1) (A1,2) (A5,5) (A4,7) (A3,7) (B8,3) (A7,3) (A9,6) (A8,4) (A91,2) (A11,10)
2. **void traverse()**: Display all persons having age < the average age of the tree in format (name, age) to the output screen by post-order traverse.
   For example, the output must be:
   (A1,2) (A8,4) (A91,2) (A7,3) (A6,1)
3. **void delete()**: Perform breadth-first traverse from the root and delete by copying the second node having age >= the average age, write the tree to the output screen by breadth-first traverse.
   For example, the output must be:
   (A6, 1) (A2, 5) (A7, 3) (A1, 2) (A4, 7) (A9, 6) (A3, 7) (A8, 4) (A91, 2)
4. **void rotateLeft()**: Perform pre-order traverse from the root and rotate the third node having non-empty right-son then rotate it to left about its right-son and display the tree to the output screen by pre-order traverse.
   For example, the output must be:
   (A6, 1) (A2, 5) (A1, 2) (A5, 5) (A4, 7) (A3, 7) (A9, 6) (A7, 3) (A8, 4) (A91, 2)