Report on Assignment-2

Properties:
1. Tree should be a BST (Binary Search Tree).
2. All nodes should be either Red or Black.
3. Root and leaf-nodes(sentinel) are Black.
4. Parent of every Red node is Black or Every Red Node has Black Children.
5. All simple paths from a node to a descendant leaf of X should have the same #of Black nodes.

Assumptions:
1. Left child to have less or equal values.
2. Right child to have greater values.
3. All nodes to have positive Integer values.
4. Any value other than 0xffffff can be inputted. (used for programming purpose)

Input Format:
1st Line : Array Representation of the Tree where each node indicates "<value><color 'R|B'>". A Null Node is represented by 'X'.
Next line contains N, the number of values to delete from the Tree.
Following N lines contains a value to delete.

Cases for Invalid Input:
1. Node to be inserted has a NULL ancestor.
2. Invalid Color.

Compiler Version:
g++ version 5.1

Reference:
RB_TRANSPLANT : CLRS Section: Section 13.4
RB_DELETE_FIXUP : CLRS Section: Section 13.4
RB_DELETE : CLRS Section: Section 13.4
LEFT_ROTATE : CLRS Section: Section 13.2
RIGHT_ROTATE : CLRS Section: Section 13.2