Read Me : Assignment 2

Input Format:

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

Compiler Version:

g++ version 5.1

Sample Test Cases:

1. **Not BST:**

20B X 22B X X 21R 10B // Right most node lesser than parent
20B 18R 21B X X 70R 22B // Left node greater than parent

Output: Invalid Tree: Tree is not a BST.

2. **Red-Red Conflict**

20B 18R 22B X 19R 21B 23R
20B 18R 22R X 19B 21R 23R

Output: Red-Red Conflict. A RED node contains a Red Child!

3. **Null Parent Insertion**

20B X 1B 2B 3B

Output: Value to be inserted has a NULL ancestor.
Invalid Input.

4. **Black Height**

10B 5B 15B 1B 6B 12B 17B
Output: 3

10B 5R 15R 1B 6B 12B 17B
Output: 2

RB Deletion related Test Cases:

1. // Deletes 5 Nodes

```
10B 5B 15B 1R 6R 12R 17R
5
17
1
6
12
15
```

Output:

```
Tree is a RB-Tree. Verified!

Deleting Node : 17
Successor node transplanted : 10B 5B 15B 1R 6R 12R X
Resultant Tree: 10B 5B 15B 1R 6R 12R X

Deleting Node : 1
Successor node transplanted : 10B 5B 15B X 6R 12R X
Resultant Tree: 10B 5B 15B X 6R 12R X

Deleting Node : 6
Successor node transplanted : 10B 5B 15B X X 12R X
Resultant Tree: 10B 5B 15B X X 12R X

Deleting Node : 12
Successor node transplanted : 10B 5B 15B X X X X
Resultant Tree: 10B 5B 15B X X X X

Deleting Node : 15
Successor node transplanted : 10B 5B X X X X X
Case 2: 10B 5R X X X X X
Resultant Tree: 10B 5R X X X X X
```

2. // Checks Case 1 & 2.

```
Output:
Tree is a RB-Tree. Verified!

Deleting Node : 30
Successor node transplanted : 32B 10B 40B 5B 20B 35B 50B 1B 6B 15B 25B X 37R 45B
60B X X X X X X X X X X 36B 38B
Case 1: 32B 10B 40B 5B 20B 37B 50B 1B 6B 15B 25B 35R 38B 45B 60B X X X X X X X
X X 36B X X
Case 2: 32B 10B 40B 5B 20B 37B 50B 1B 6B 15B 25B 35R 38B 45B 60B X X X X X X X
X X 36R X X
Resultant Tree: 32B 10B 40B 5B 20B 37B 50B 1B 6B 15B 25B 35B 38B 45B 60B X X X
X X X X X X 36R X X
```

3. // Checks case 2 & 2.

```
30B 10R 40R 5B 20B 35B 50B 1B 6B 15B 25B 32B 37B 45B 60B
1
30

Output:
Tree is a RB-Tree. Verified!

Deleting Node : 30
Successor node transplanted : 32B 10R 40R 5B 20B 35B 50B 1B 6B 15B 25B X 37B 45B
60B
Case 2: 32B 10R 40R 5B 20B 35B 50B 1B 6B 15B 25B X 37R 45B 60B
Case 2: 32B 10R 40R 5B 20B 35B 50R 1B 6B 15B 25B X 37R 45B 60B
Resultant Tree: 32B 10R 40B 5B 20B 35B 50R 1B 6B 15B 25B X 37R 45B 60B
```

4. // Checks Case 2, 3 & 4.

```
30B 10B 40B 5B 20B 35B 50B 1B 6B 15B 25B 32B 37B 45R 55B X X X X X X X X X X
X X 44B 46B X X
1
30
```

```
Output:
Tree is a RB-Tree. Verified!

Deleting Node : 30
Successor node transplanted : 32B 10B 40B 5B 20B 35B 50B 1B 6B 15B 25B X 37B 45R
55B X X X X X X X X X X X X 44B 46B
Case 2: 32B 10B 40B 5B 20B 35B 50B 1B 6B 15B 25B X 37R 45R 55B X X X X X X X X
X X X X 44B 46B
Case 3: 32B 10B 40B 5B 20B 35B 45B 1B 6B 15B 25B X 37R 44B 50R X X X X X X X X
X X X X X X
Case 4: 32B 10B 45B 5B 20B 40B 50B 1B 6B 15B 25B 35B 44B 46B 55B X X X X X X X
X X 37R X X X X
Resultant Tree: 32B 10B 45B 5B 20B 40B 50B 1B 6B 15B 25B 35B 44B 46B 55B X X X
X X X X X X 37R X X X X
```