BST Insertion

Problem Statement

Implement the following two functions to insert a node nde into a BST rooted at r.

```
void BST_insert_after(node *r, int pos, node *nde);
void BST_insert_before(node *r, int pos, node *nde);
```

Assume the following declaration of the node struct.
(and do not redeclare it in your submitted program code)

```
struct node {
    char c;
    int pri, l_sz, r_sz;
    // pri, size of left child and right child
    node *lc, *rc, *p;
    // pointer to left child, right child, parent
};
```

Description:

- Regard the elements stored in the BST as an ordered sequence.
- The function insert_before will insert the node nde before the pos -th element in the sequence. That is, the node nde will be inserted between the (pos-1) -th element and the pos -th element in the BST.
- Similarly, the function insert_after will insert the node nde after the pos -th element in the BST.

Requirements:

- The functions should run in $O(\log n)$ time, where n is the number of nodes in the BST.
- You need to adjust the pointers and auxiliary fields to maintain the BST structure and property after insertion.
- You may assume that pos is a valid index between 1 and n , where n is the number of nodes in the BST rooted at r .

Note that, if a pointer does not reference any node, it should be set to be `NULL` or `nullptr`.

Submission Instructions

This is a function implementation task. Your submitted code must include the following identifier:

```
/* probID: W9-A2-Insertion */
```

and must include the implementation of the `BST_insert_before` and `BST_insert_after` function (additional function declarations/implementations are allowed if necessary), but must not include a `main` function, as its presence will cause compilation errors.

When submitting, choose the language `C++ - function only`.