2) DELETE A NODE IN THE BST:

BEST METHOD :
We first find the parent of the node which has to be deleted.
Then we take the left node of the node of the node to be deleted into the temp varaible.
Then we point the parent ->left=parent->left->left.
Then we insert the temp into the binary tree.
Time Complexity : O(N)

CODE OF THE ABOVE METHOD:

```
void Insert(TreeNode *root,TreeNode *toBeInserted){
        if(toBeInserted->val<root->val){
            if(root->left==NULL){
                root->left=toBeInserted;
                return;
            }
            else{
                Insert(root->left,toBeInserted);
            }
        }
        else{
            if(root->right==NULL){
                root->right=toBeInserted;
                return;
            }
            else{
                Insert(root->right,toBeInserted);
            }
        }
    }
    TreeNode* deleteNode(TreeNode* root, int key) {
        if(root==NULL){
            return NULL;
        }
        if(root->val==key){
            if(root->left==NULL && root->right==NULL){
                delete(root);
                return NULL;
            }
            else if(root->left==NULL && root->right!=NULL){
                TreeNode *head=root->right;
                delete(root);
                return head;
            }
            else if(root->left!=NULL && root->right==NULL){
                TreeNode *head=root->left;
                delete(root);
                return head;
        }
        else{
            TreeNode *head=root->left;
            TreeNode *toBeInserted=head->right;
            head->right=root->right;
            delete(root);
            if(toBeInserted!=NULL){
                Insert(head,toBeInserted);
            }
            return head;
        }
    }
    else if(key<root->val){
        root->left=deleteNode(root->left,key);
        return root;
    }
    else{
        root->right=deleteNode(root->right,key);
        return root;
    }
}
```