**Predecessor And Successor In BST**

void traverse(BinaryTreeNode<int>\* root, vector<int> &ans){

    if(root==NULL) return;

    traverse(root->left,ans);

    ans.push\_back(root->data);

    traverse(root->right,ans);

}

pair<int,int> predecessorSuccessor(BinaryTreeNode<int>\* root, int key)

{

    // Write your code here.

    vector<int> ans;

    traverse(root,ans);

    int i;

    for(i=0; i<ans.size(); i++){

        if(ans[i]==key) break;

    }

    pair<int,int> res;

    res.first = -1;

    res.second =  -1;

    int n = ans.size();

    if(n==1) return res;

    if(i==0 && n>1)

     {

       res.second = ans[i+1];

       return res;

     }

    if(i==ans.size()-1 && n>1)

       {

         res.first = ans[i-1];

         return res;

       }

         res.first = ans[i-1];

         res.second = ans[i+1];

         return res;

}