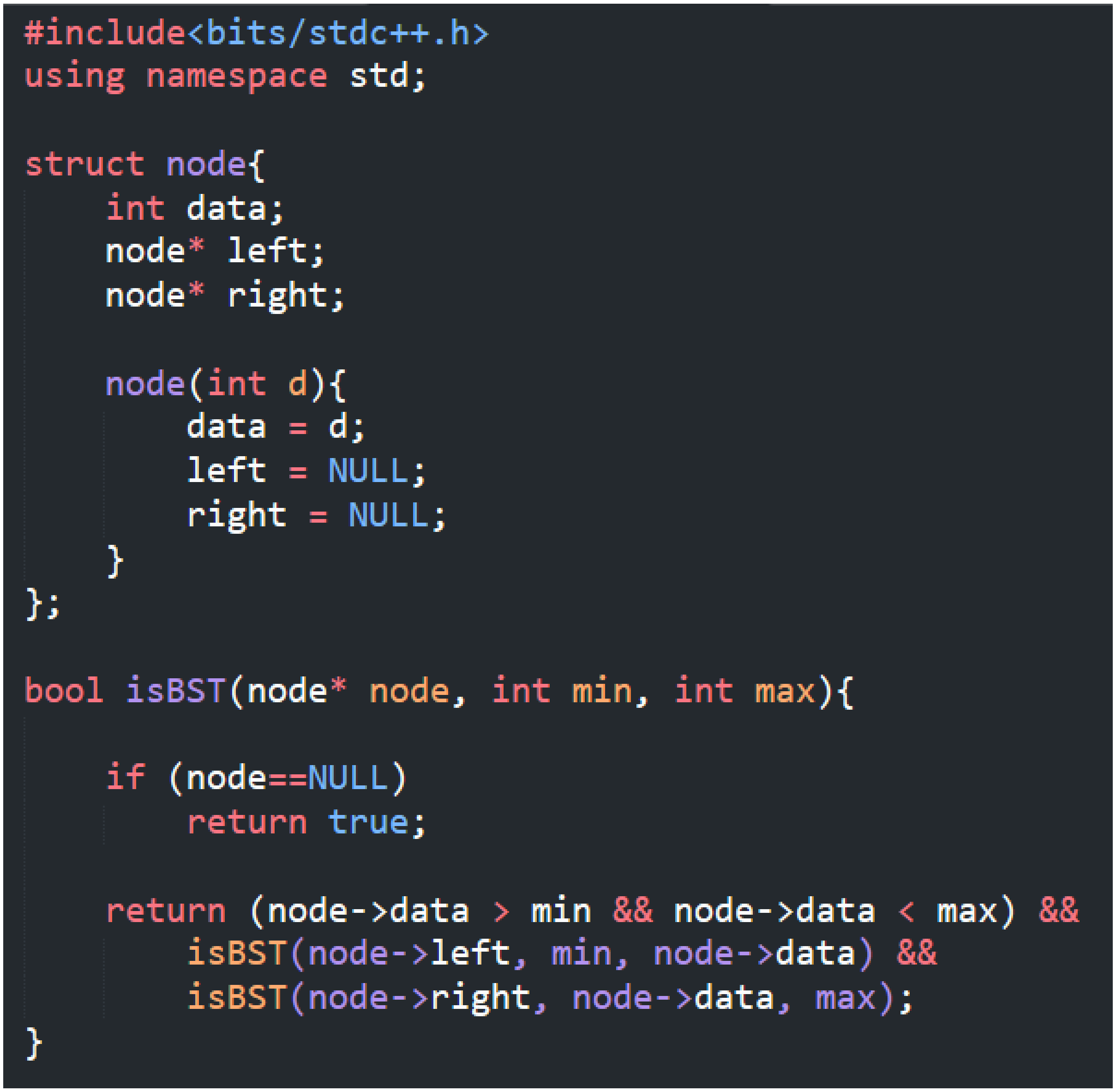
IT DS 201 LAB

SUBMITTED BY ADITYA SINGH 2K19/EP/005

Program 17 : Write a program to check whether the given tree is a Binary Search Tree or not.

CODE



```
#include<bits/stdc++.h>
using namespace std;

struct node{
    int data;
    node* left;
    node* right;

    node(int d){
        data = d;
        left = NULL;
        right = NULL;
    }
};

bool isBST(node* node, int min, int max){

    if (node==NULL)
        return true;

    return (node->data > min && node->data < max) &&
        isBST(node->left, min, node->data) &&
        isBST(node->right, node->data, max);
}
```

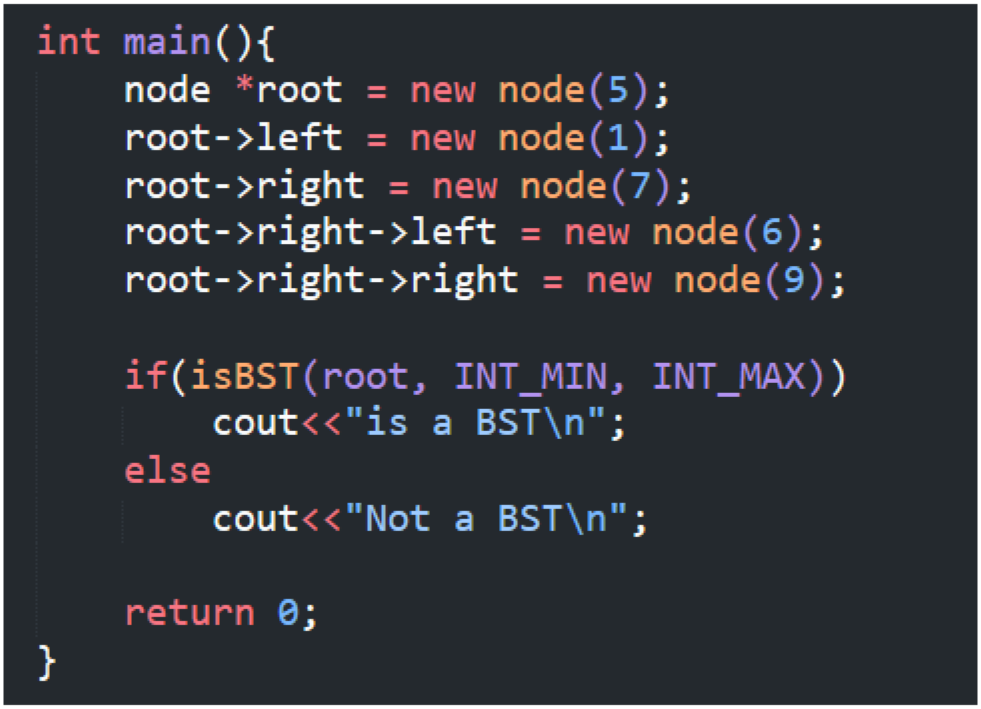
ALGORITHM

Traverse down the tree keeping track of the narrowing min and max allowed values as it goes, looking at each node only once. The condition we need to check at each node is:

a) If the node is the left child of its parent, it must be smaller than (or equal to) the parent, and it must pass down the value from its parent to its right subtree to make sure none of the nodes in that subtree is greater than the parent.

b) If the node is the right child of its parent, it must be larger than the parent, and it must pass down the value from its parent to its left subtree to make sure none of the nodes in that subtree is lesser than the parent.

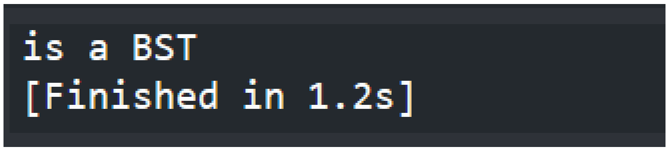
INPUT/OUTPUT



```
int main(){
    node *root = new node(5);
    root->left = new node(1);
    root->right = new node(7);
    root->right->left = new node(6);
    root->right->right = new node(9);

    if(isBST(root, INT_MIN, INT_MAX))
        cout<<"is a BST\n";
    else
        cout<<"Not a BST\n";

    return 0;
}
```



```
is a BST
[Finished in 1.2s]
```

END