LAB TASK # 08

MOHAMMAD BASIL ALI KHAN

20K-0477

Task#01:

- **Code**

```cpp
#include<iostream>
#define INT_MIN (-2147483647 - 1)
#define INT_MAX 2147483647

using namespace std;

class Node{
    public:
        int data;
        Node *left;
        Node *right;

        Node()
        {
            left = NULL;
            right = NULL;
            data = 0;
        }

        Node(int val){
            left = NULL;
            right = NULL;
            data = val;
        }
        int isBST(Node* Node)
        {
            return(isBSTUtil(Node, INT_MIN, INT_MAX));
        }
        int isBSTUtil(Node* Node, int min, int max)
        {
            if(Node==NULL)
                return 1;
            if (Node->data < min || Node->data > max)
```

```cpp
        {
            return(isBSTUtil(Node, INT_MIN, INT_MAX));
        }
        int isBSTUtil(Node* Node, int min, int max)
        {
            if(Node==NULL)
                return 1;
            if (Node->data < min || Node->data > max)
                return 0;

            return
            isBSTUtil(Node->left, min, Node->data-1) && isBSTUtil(Node->right, Node->data+1, max);
        }
};
int main(){
    Node obj;
    Node *root = new Node(40);
    root->left = new Node(9);
    root->right = new Node(11);
    root->left->left = new Node(1);
    root->left->right = new Node(30);

    if(obj.isBST(root)){
        cout << "Is BST" << endl;
        cout << "10 Marks Assigned";
    }
    else{
        cout << "Not a BST" << endl;
        cout << "0 Marks Assigned";
    }
}
```

- **Output**

Task#02:

- **Code**

```cpp
#include<iostream>

using namespace std;

class Node{
    public:
        int data;

        Node *left, *right;

        Node(int val)
        {
            data = val;
            left = NULL;
            right = NULL;
        }


};

Node *insertBST(Node *key, int val)
{
    if(key==NULL)
    {
        return new Node(val);
    }
    if(val < key->data)
    {
        key->left = insertBST(key->left, val);
    }
    else
    {
        key->right = insertBST(key->right, val);
```

```cpp
        }
        else
        {
            key->right = insertBST(key->right, val);
        }
        return key;
}

void Display(Node *key)
{
    if(key==NULL)
    {
        return;
    }
    Display(key->left);
    cout << key->data << " ";
    Display(key->right);
    cout << key->data << " ";
}

int main()
{
    Node *key = NULL;
    key = insertBST(key, 15);
    insertBST(key, 10);
    insertBST(key, 20);
    insertBST(key, 8);
    insertBST(key, 12);
    insertBST(key, 16);
    insertBST(key, 25);
    Display(key);
}
```

- **Output**