**Additional Program**

**Check if a Tree is Binary Search Tree**

#include <stdio.h>

#include <stdlib.h>

 struct node

{

    int data;

    struct node\* left;

    struct node\* right;

};

 static struct node \*prev = NULL;

 /\*Function to check whether the tree is BST or not\*/

int is\_bst(struct node\* root)

{

    if (root)

    {

        if (!is\_bst(root->left)) //moves towards the leftmost child of the tree and checks for the BST

            return 0;

        if (prev != NULL && root->data <= prev->data)

            return 0;

        prev = root;

        return is\_bst(root->right);    //moves the corresponding right child of the tree and checks for the BST

    }

    return 1;

}

struct node\* newNode(int data)

{

    struct node\* node = (struct node\*)malloc(sizeof(struct node));

    node->data = data;

    node->left = NULL;

    node->right = NULL;

     return(node);

}

 int main()

{

  /\*

    The input tree is as shown below

                40

                / \

            20        60

            / \       \

        10        30      80

                          \

                            90

  \*/

    struct node \*root = newNode(40);

    root->left        = newNode(20);

    root->right       = newNode(60);

    root->left->left  = newNode(10);

    root->left->right = newNode(30);

    root->right->right = newNode(80);

    root->right->right->right = newNode(90);

    if (is\_bst(root))

        printf("TREE 1 Is BST");

    else

        printf("TREE 1 Not a BST");

    prev = NULL;

/\*

    The input tree is as shown below

                50

                / \

            20        30

            / \

        70        80

        / \          \

    10     40     60

\*/

    struct node \*root1 = newNode(50);

    root1->left  = newNode(20);

    root1->right  = newNode(30);

    root1->left->left  = newNode(70);

    root1->left->right = newNode(80);

    root1->left->left->right = newNode(40);

    root1->left->left->left= newNode(90);

    if (is\_bst(root1))

        printf("TREE 2 Is BST");

    else

        printf("TREE 2 Not a BST");

    return 0;

}