**Binary search tree has a unique property, that is the left sub tree has the value less than that of the root and the right subtree hold the value greater than that of the tree**

***CODE:***

#include<stdio.h>

#include<stdlib.h>

struct node **//create a node structure**

{

int key;

struct node \*left, \*right;

};

struct node \*newNode(int item) { **//Function to create a new node for every inserttion**

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

temp->key = item; **//Set the data to be inserted as the data feild**

temp->left = temp->right = NULL; **//Setting the left and right field as null**

return temp;

}

struct node\* insert(struct node\* node, int key)

{

**/\* If the tree is empty, return a new node \*/**

if (node == NULL) return newNode(key);

**/\* Otherwise, recur down the tree \*/**

if (key < node->key)

node->left = insert(node->left, key);

else if (key > node->key)

node->right = insert(node->right, key);

**/\* return the (unchanged) node pointer \*/**

return node;

}

int main()

{

struct node \*root = NULL;

root = insert(root, 50);

insert(root, 30);

insert(root, 20);

insert(root, 40);

insert(root, 70);

insert(root, 60);

insert(root, 80);

return 0;

}