

# Experiment 10: Write a program to implement a Binary Search Tree.

Binary Search Tree is a node-based binary tree data structure which has the following properties:-

- The left subtree of a node contains only nodes with keys lesser than the node's key.
- The right subtree of a node contains only nodes with keys greater than the node's key
- The left and right subtree each must also be a binary search tree

```
#include<stdio.h>
#include<conio.h>
struct node
{
    int key;
    struct node *left, *right;
};
struct node *newNode(int item)
{
    struct node *temp = (struct node *)malloc(sizeof(struct node));
    temp->key = item;
    temp->left = temp->right = NULL;
    return temp;
}
void inorder(struct node *root)
{
    if (root != NULL)
    {
        inorder(root->left);
        printf("%d \n", root->key);
        inorder(root->right);
    }
}
```

```

struct node* insert(struct node* node, int key)
{
    if (node == NULL) return newNode(key);
    if (key < node->key)
        node->left = insert(node->left, key);
    else if (key > node->key)
        node->right = insert(node->right, key);
    return node;
}

int main()
{
    struct node *root = NULL;
    root = insert(root, 60);
    insert(root, 10);
    insert(root, 20);
    insert(root, 30);
    insert(root, 40);
    insert(root, 50);
    insert(root, 60);
    insert(root, 70);
    inorder(root);
    return 0;
}

```

