//C program to demonstrate insert operation in binary tree//

#include<stdio.h>

#include<stdlib.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 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, 5);

insert(root, 3);

insert(root, 2);

insert(root, 4);

insert(root, 7);

insert(root, 6);

insert(root, 8);

// print inoder traversal of the BST

inorder(root);

return 0;

}

Output:

2

3

4

5

6

7

8