Practical no 7 1.To study and implement binary search tree (BST) .Write a function form Inorder, Preorder and Postorder traversal on a BST.

CODE:

#include<stdio.h>

#include<stdlib.h>

struct node {

    struct node\* left;

    struct node\* right;

    int data;

};

struct node\*createNode(){

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

    return root;

}

void inorderTrav(struct node\* root){

    if(root){

    inorderTrav(root->left);

    printf("Inorder data is %d\n",root->data);

    inorderTrav(root->right);

    }

}

void preorderTrav(struct node\* root){

    if(root){

    printf("Preorder data is %d\n",root->data);

    preorderTrav(root->left);

    preorderTrav(root->right);

    }

}

void postorderTrav(struct node\* root){

    if(root){

    postorderTrav(root->left);

    postorderTrav(root->right);

    printf("Preorder data is %d\n",root->data);

    }

}

struct node\*insertR(struct node\*root,int key){

    if(root == NULL){

        root = createNode();

        if (root == NULL){

            return root;

        }

        root->data = key;

        root->left = NULL;

        root->right = NULL;

    }

    else if(key < root->data){

        root->left = insertR(root->left,key);

    }else if(key > root->data){

        root->right = insertR(root->right,key);

    }

    return root;

}

int main(){

    struct node\*root;

    root = insertR(root,1);

    root = insertR(root,4);

    root = insertR(root,5);

    root = insertR(root,2);

    inorderTrav(root);

}

OUTPUT:

Inorder data is 1

Inorder data is 2

Inorder data is 4

Inorder data is 5