**Practical-8**

**AIM:**

**Write a program which create binary search tree and Implement tree traversing methods inorder, preorder and post-order traversal.**

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

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*left, \*right;

}\*root=NULL;

void inorder(struct node \*root)

{

if (root != NULL)

{

inorder(root->left);

printf("%d ", root->data);

inorder(root->right);

}

}

void preorder(struct node \*root)

{

if (root != NULL)

{

printf("%d ", root->data);

preorder(root->left);

preorder(root->right);

}

}

void postorder(struct node \*root)

{

if (root != NULL)

{

postorder(root->left);

postorder(root->right);

printf("%d ", root->data);

}

}

void CreateBST(struct node\* node, int val)

{

if (root == NULL)

{

struct node \*newnode = (struct node \*)malloc(sizeof(struct node)); newnode->data = val;

newnode->left = newnode->right = NULL;

root=newnode;

printf("%d is Root Node",val);

}

else if (val < node->data)

{

if(node->left==NULL)

{

struct node \*newnode = (struct node \*)malloc(sizeof(struct node)); newnode->data = val;

newnode->left = newnode->right = NULL;

node->left=newnode;

printf("%d is Left Child of %d",val,node->data);

}

else

{

CreateBST(node->left,val);

}

}

else if (val >= node->data)

{

if(node->right==NULL)

{

struct node \*newnode = (struct node \*)malloc(sizeof(struct node)); newnode->data = val

newnode->left = newnode->right = NULL; node->right=newnode;

printf("%d is Right Child of %d",val,node->data); }

else

{

CreateBST(node->right,val);

}

}

}

void main()

{

int val;

while(1)

{

printf("\nEnter Node : ");

scanf("%d",&val);

if(val==99)

break;

CreateBST(root, val);

}

printf("\nInorder :");

inorder(root);

printf("\nPreorder :");

preorder(root);

printf("\nPostorder :");

postorder(root);

getch();

}