LAB 8

Demonstrate Binary Search Tree

# include <stdio.h>

# include<stdlib.h>

struct node

{

int data;

struct node\*left;

struct node\*right;

};

struct node\*root=NULL;

void create()

{

struct node\*new\_node = (struct node\*)malloc(sizeof(struct node));

int val;

printf("Enter data:");

scanf("%d",&val);

new\_node->data=val;

new\_node->left=NULL;

new\_node->right=NULL;

if (root==NULL)

{

root=new\_node;

}

else

{

struct node\*ptr=root;

struct node\*ptr1=NULL;

while(ptr!=NULL)

{

ptr1=ptr;

if(val<ptr->data)

{

ptr=ptr->left;

}

else

{

ptr=ptr->right;

}

}

if(val<ptr1->data)

ptr1->left=new\_node;

else

ptr1->right=new\_node;

}

}

void postorder(struct node\* ptr)

{

if(ptr!=NULL)

{

postorder(ptr->left);

postorder(ptr->right);

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

}

}

void preorder(struct node\* ptr)

{

if(ptr!=NULL)

{

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

preorder(ptr->left);

preorder(ptr->right);

}

}

void inorder(struct node\* ptr)

{

if(ptr!=NULL)

{

inorder(ptr->left);

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

inorder(ptr->right);

}

}

void main()

{

int n;

printf("Enter no of nodes to be inserted:");

scanf("%d",&n);

for(int i=0;i<n;i++)

{

create();

}

printf("\nInorder:\n");

inorder(root);

printf("\nPreorder:\n");

preorder(root);

printf("\nPostorder:\n");

postorder(root);

}

OUTPUT:

