**Q. WAP for Binary search tree.**

#include <stdio.h>

#include<stdlib.h>

int x;

struct Node

{

int data;

struct Node \*left;

struct Node \*right;

};

struct Node \*CreateNode(int x)

{

struct Node \*newNode;

newNode=malloc(sizeof(struct Node));

newNode->data=x;

newNode->left=NULL;

newNode->right=NULL;

return newNode;

}

struct Node \*insert (struct Node \*root, int x)

{

if (root==NULL)

{

return CreateNode(x);

}

else if(x>root->data)

{

root->right=insert(root->right,x);

}

else

{

root->left=insert(root->left,x);

}

return root;

}

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);

inorder(root->left);

inorder(root->right);

}

}

void postorder(struct Node\*root)

{

if (root!=NULL)

{

inorder(root->left);

inorder(root->right);

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

}

}

int main()

{

struct Node \*root;

printf("Enter value at root node : ");

scanf("%d",&x);

root = CreateNode(x);

printf("Insert any value : ");

scanf("%d",&x);

insert(root,x);

printf("Insert any value : ");

scanf("%d",&x);

insert(root,x);

printf("Inorder : ");

inorder(root);

printf("\nPreorder : ");

preorder(root);

printf("\nPostorder : ");

postorder(root);

}

**OUTPUT :**

