using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

class Node

{

public int Data;

public Node Lptr;

public Node Rptr;

public void Display()

{

Console.WriteLine(Data + "==>");

}

}

class BST

{

public Node root;

public BST()

{

root = null;

}

public void Insert(int data)

{

Node node = new Node();

node.Data = data;

if (root == null)

{

root = new Node();

root.Data = data;

Console.WriteLine("Root Node Added");

}

else

{

Node ptr = root;

Node parent;

while (true)

{

parent = ptr;

if (data < ptr.Data)

{

ptr = ptr.Lptr;

if (ptr == null)

{

parent.Lptr = node;

Console.WriteLine("Node added at left");

break;

}

}

else

{

ptr = ptr.Rptr;

if (ptr == null)

{

parent.Rptr = node;

Console.WriteLine("Node added at right");

break;

}

}

}

}

}

public void InOrderTraversal(Node parent)

{

if(parent!=null)

{

InOrderTraversal(parent.Lptr);

Console.WriteLine(parent.Data);

InOrderTraversal(parent.Rptr);

}

}

public void PreOrderTraversal(Node parent)

{

if (parent != null)

{

Console.WriteLine(parent.Data);

PreOrderTraversal(parent.Lptr);

PreOrderTraversal(parent.Rptr);

}

}

public void PostOrderTraversal(Node parent)

{

if (parent != null)

{

PostOrderTraversal(parent.Lptr);

PostOrderTraversal(parent.Rptr);

Console.WriteLine(parent.Data);

}

}

}

class Program

{

static void Main(string[] args)

{

BST bst = new BST();

bst.Insert(10);

bst.Insert(20);

bst.Insert(30);

bst.Insert(1);

Console.WriteLine("InOrder");

bst.InOrderTraversal(bst.root);

Console.WriteLine("PreOrder");

bst.PreOrderTraversal(bst.root);

Console.WriteLine("PostOrder");

bst.PostOrderTraversal(bst.root);

//Console.WriteLine("After Sorting array : ");

}

}