NAME – SHRAVANI BAHULEKAR

ROLL N0 – 32

PRN – 12111224

ASSIGNMENT 2

BATCH – D2

BFS –

import java.util.LinkedList;  
import java.util.Queue;  
  
class TreeNode {  
 int val;  
 TreeNode left;  
 TreeNode right;  
  
 TreeNode(int val) {  
 this.val = val;  
 left = null;  
 right = null;  
 }  
}  
  
public class BFS {  
 public static void bfsTraversal(TreeNode root) {  
 if (root == null) {  
 return;  
 }  
  
 Queue<TreeNode> queue = new LinkedList<>();  
 queue.add(root);  
  
 while (!queue.isEmpty()) {  
 TreeNode current = queue.poll();  
 System.*out*.print(current.val + " ");  
  
 if (current.left != null) {  
 queue.add(current.left);  
 }  
  
 if (current.right != null) {  
 queue.add(current.right);  
 }  
 }  
 }  
  
 public static void main(String[] args) {  
 // Construct a sample binary tree  
 TreeNode root = new TreeNode(1);  
 root.left = new TreeNode(2);  
 root.right = new TreeNode(3);  
 root.left.left = new TreeNode(4);  
 root.left.right = new TreeNode(5);  
 root.right.left = new TreeNode(6);  
 root.right.right = new TreeNode(7);  
 root.right.right.left = new TreeNode(8);  
 root.right.right.left.right = new TreeNode(9);  
  
 System.*out*.print("BFS Traversal of the binary tree: ");  
 *bfsTraversal*(root);  
 }  
}

DFS –

import java.util.LinkedList;  
import java.util.Queue;  
  
class Node {  
 int key;  
 Node left, right;  
  
 public Node(int item)  
 {  
 key = item;  
 left = right = null;  
 }  
}  
  
class BinaryTree {  
 Node root;  
  
 BinaryTree() { root = null; }  
  
 void printInorder(Node node)  
 {  
 if (node == null)  
 return;  
  
 printInorder(node.left);  
  
 System.*out*.print(node.key + " ");  
  
 printInorder(node.right);  
 }  
  
 void bfsTraversal(Node root) {  
 if (root == null) {  
 return;  
 }  
  
 Queue<Node> queue = new LinkedList<>();  
 queue.add(root);  
  
 while (!queue.isEmpty()) {  
 Node current = queue.poll();  
 System.*out*.print(current.key + " ");  
  
 if (current.left != null) {  
 queue.add(current.left);  
 }  
  
 if (current.right != null) {  
 queue.add(current.right);  
 }  
 }  
 }  
  
 public static void main(String[] args)  
 {  
 BinaryTree tree = new BinaryTree();  
 tree.root = new Node(1);  
 tree.root.left = new Node(2);  
 tree.root.right = new Node(3);  
 tree.root.left.left = new Node(4);  
 tree.root.left.right = new Node(5);  
 tree.root.right.left = new Node(6);  
 tree.root.right.right = new Node(7);  
 tree.root.right.right.left = new Node(8);  
 tree.root.right.right.left.right = new Node(9);  
  
 System.*out*.println("\nDFS traversal of binary tree is ");  
 tree.printInorder(tree.root);  
  
 System.*out*.println("\nBFS traversal of binary tree is ");  
 tree.bfsTraversal(tree.root);  
 }  
}