import java.util.\*;

class GraphNode {

int value;

List<GraphNode> neighbors;

GraphNode(int value) {

this.value = value;

neighbors = new ArrayList<>();

}

}

public class BFS\_DFS\_2 {

public static void bfs(GraphNode start) {

if (start == null) {

return;

}

Queue<GraphNode> queue = new LinkedList<>();

Set<GraphNode> visited = new HashSet<>();

queue.add(start);

visited.add(start);

while (!queue.isEmpty()) {

GraphNode current = queue.poll();

System.out.print(current.value + " ");

for (GraphNode neighbor : current.neighbors) {

if (!visited.contains(neighbor)) {

queue.add(neighbor);

visited.add(neighbor);

}

}

}

}

public static void dfs(GraphNode start) {

if (start == null) {

return;

}

Stack<GraphNode> stack = new Stack<>();

Set<GraphNode> visited = new HashSet<>();

stack.push(start);

while (!stack.isEmpty()) {

GraphNode current = stack.pop();

if (!visited.contains(current)) {

System.out.print(current.value + " ");

visited.add(current);

for (GraphNode neighbor : current.neighbors) {

stack.push(neighbor);

}

}

}

}

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

GraphNode node40 =new GraphNode(40);

GraphNode node10 =new GraphNode(10);

GraphNode node20 =new GraphNode(20);

GraphNode node30 =new GraphNode(30);

GraphNode node60 =new GraphNode(60);

GraphNode node50 =new GraphNode(50);

GraphNode node70 =new GraphNode(70);

node40.neighbors.add(node10);

node40.neighbors.add(node20);

node10.neighbors.add(node30);

node20.neighbors.add(node10);

node20.neighbors.add(node30);

node20.neighbors.add(node60);

node20.neighbors.add(node50);

node30.neighbors.add(node60);

node60.neighbors.add(node70);

node50.neighbors.add(node70);

System.out.println("Choose traversal:");

System.out.println("1. BFS");

System.out.println("2. DFS");

int choice=sc.nextInt();

GraphNode startGraphNode = node40;

switch (choice) {

case 1:

System.out.print("BFS traversal of the graph: ");

bfs(startGraphNode);

break;

case 2:

System.out.print("DFS traversal of the graph: ");

dfs(startGraphNode);

break;

default:

System.out.println("Invalid choice");

break;

}

}

}