**package** GraphTheory;

**import** java.util.LinkedList;

**import** java.util.Queue;

**import** java.util.Scanner;

**public class** BreathFastSearchGraph {

**public static void** BFS(**int** vertex, **int** edge,**int**[][] graph)

{

Scanner input = **new** Scanner(System.***in***);

Queue< Integer > queue = **new** LinkedList<>();

**int**[] distance = **new int**[vertex];

**int**[] parent = **new int**[vertex];

**boolean**[] visited = **new boolean**[vertex];

**for** (**int** i=0;i<vertex;i++)

{

visited[i] = **false**;

parent[i] = -1;

distance[i] = 0;

}

**for** (**int** i=0;i<vertex;i++)

{

**for** (**int** j=0;j<vertex;j++)

{

graph[i][j] = 0;

}

}

**for** (**int** i =0; i<edge;i++)

{

**int** n1 = input.nextInt();

**int** n2 = input.nextInt();

graph[n1][n2] = 1;

graph[n2][n1] = 1;

}

queue.add(0);

visited[0] = **true**;

**while** (!queue.isEmpty())

{

**int** j = queue.remove();

System.***out***.print(j+**" "**);

**for** (**int** i=0;i<vertex;i++)

{

**if** (graph[j][i] ==1 && !visited[i])

{

distance[i] = distance[j] + 1;

parent[i] = j;

queue.add(i);

visited[i] = **true**;

}

}

}

**for** (**int** i=0;i<vertex;i++)

{

System.***out***.println(**"Distance "**+i+**": "**+distance[i]);

}

System.***out***.println(**"------------------"**);

**for** (**int** i=0;i<vertex;i++)

{

System.***out***.println(**"Parent "**+i+**": "**+parent[i]);

}

}

**public static void** main(String[] args) {

**int**[][] graph = **new int**[100][100];

Scanner input = **new** Scanner(System.***in***);

System.***out***.println(**"Enter number of vertex: "**);

**int** vertex = input.nextInt();

System.***out***.println(**"Enter number of edges: "**);

**int** edge = input.nextInt();

*BFS*(vertex,edge,graph);

}

}