**DATA STRUCTURES AND ALGORIHTMS**

**LAB ASSIGNMENT - 10**

NAME – KAPAROTU VENKATA SURYA THARANI

COURSE – AIDE

SECTION - “A”

USN ID – 22BTRAD018

Q – Implement Graph Data Structure using Java.

CODE:

import java.util.\*;

class Graph {

private int V; // No. of vertices

private LinkedList<Integer> adj[]; // Adjacency List

// Constructor

Graph(int v) {

V = v;

adj = new LinkedList[v];

for (int i=0; i<v; ++i)

adj[i] = new LinkedList();

}

// Function to add an edge into the graph

void addEdge(int v, int w) {

adj[v].add(w);

}

// Prints BFS traversal from a given source s

void BFS(int s) {

// Mark all the vertices as not visited(By default set as false)

boolean visited[] = new boolean[V];

// Create a queue for BFS

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

// Mark the current node as visited and enqueue it

visited[s]=true;

queue.add(s);

while (queue.size() != 0) {

// Dequeue a vertex from queue and print it

s = queue.poll();

System.out.print(s+" ");

// Get all adjacent vertices of the dequeued vertex s

// If a adjacent has not been visited, then mark it visited and enqueue it

Iterator<Integer> i = adj[s].listIterator();

while (i.hasNext()) {

int n = i.next();

if (!visited[n]) {

visited[n] = true;

queue.add(n);

}

}

}

}

}

OUTPUT:

