BFS ALGORITHM

```
import java.util.*;
public class bfs {
private int V;
private LinkedList<Integer> adj[];
bfs(int v) {
V = v;
adj = new LinkedList[v];
for (int i = 0; i < v; ++i)
adj[i] = new LinkedList();
 }
 void addEdge(int v, int w) {
 adj[v].add(w);
  }
 void BFS(int s) {
 boolean visited[] = new boolean[V];
 LinkedList<Integer> queue = new LinkedList();
 visited[s] = true;
 queue.add(s);
 while (queue.size() != 0) {
  s = queue.poll();
  System.out.print(s + " ");
  Iterator<Integer> i = adj[s].listIterator();
  while (i.hasNext()) {
```

```
  int n = i.next();
  if (!visited[n]) {
  visited[n] = true;
        queue.add(n);
       }
     }
   }
  public static void main(String args[]) {
    bfs g = new bfs(4);
    g.addEdge(0, 1);
    g.addEdge(0, 2);
    g.addEdge(1, 2);
    g.addEdge(2, 0);
    g.addEdge(2, 3);
    g.addEdge(3, 3);
    System.out.println("Following is Breadth First Traversal " + "(starting from
vertex 2)");
    g.BFS(2);
  }
}
```

OUTPUT: