Task 5

Breadth-First Search (BFS) Implementation

For a given undirected graph, implement BFS to traverse the graph starting from a given node and print each node in the order it is visited.

ANS:

```
package Day14;
import java.util.*;
public class BFS {
  public static void bfs(Map<Integer, List<Integer>> graph, int start) {
     boolean[] visited = new boolean[graph.size()]; // Array to track
visited nodes
     Queue<Integer> queue = new LinkedList<>();    // Queue for BFS
     visited[start] = true; // Mark the start node as visited
     queue.add(start);      // Add the start node to the queue
     while (!queue.isEmpty()) {
        int node = queue.poll();  // Remove a node from the queue
        System.out.println(node); // Process the node (print it in this
case)
        // Iterate through all adjacent nodes
        for (int neighbor : graph.get(node)) {
           if (!visited[neighbor]) {
              visited[neighbor] = true;  // Mark the neighbor as visited
              queue.add(neighbor);       // Add the neighbor to the queue
           }
        }
     }
  }
  public static void main(String[] args) {
     // Example graph represented as an adjacency list
     Map<Integer, List<Integer>> graph = new HashMap<>();
     graph.put(0, Arrays.asList(1, 2));
     graph.put(1, Arrays.asList(0, 3, 4));
     graph.put(2, Arrays.asList(0, 5));
     graph.put(3, Arrays.asList(1));
     graph.put(4, Arrays.asList(1, 5));
     graph.put(5, Arrays.asList(2, 4));
```

```
        // Perform BFS starting from node 0
        bfs(graph, 0);
    }
}
```