**Task 6**

**Depth-First Search (DFS) Recursive**

**Write a recursive DFS function for a given undirected graph. The function should visit every node and print it out.**

**ANS:**

```
package Day14;
import java.util.*;
public class DFS {
  // Recursive DFS function
  public static void dfs(Map<Integer, List<Integer>> graph, int node,
boolean[] visited) {
      // Mark the current node as visited
      visited[node] = true;
      // Print the current node
      System.out.println(node);
      // Recur for all adjacent nodes
      for (int neighbor : graph.get(node)) {
          if (!visited[neighbor]) {
              dfs(graph, neighbor, visited);
          }
      }
  }
  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));
      int numNodes = graph.size();
      boolean[] visited = new boolean[numNodes]; // Array to keep track
of visited nodes
      // Perform DFS starting from node 0
      dfs(graph, 0, visited);
  }
}
```