**Infinite Champions Programme – Day 3 (Assignment Sheet)**

**Instructions  
• Deadline: Submit your solutions by 29th September, 2025, EOD.  
• Platform: Test your solutions on LeetCode  
• Collaboration: Discussing concepts is encouraged, but all code must be your own.**

1. [Search a 2D Matrix](https://leetcode.com/problems/search-a-2d-matrix/)  
   • **Problem**: You are given an m × n matrix with the following properties:
   1. Each row is sorted in ascending order.
   2. The first integer of each row is greater than the last integer of the previous row.  
      Given a target integer, return true if it exists in the matrix, otherwise false.

• **Objective**: Use binary search by treating the 2D matrix as a 1D sorted array (index mapping).  
• **YouTube Solution (Java)**: [Search a 2D Matrix – Java Solution](https://www.youtube.com/watch?v=eT0UqrYuqbg)

1. [Spiral Matrix](https://leetcode.com/problems/spiral-matrix/)  
   • **Problem**: Given an m × n matrix, return all elements in **spiral order** (clockwise). You must traverse the matrix layer by layer, shrinking the boundaries after each pass.  
   • **Objective**: Implement a solution using four boundary variables (top, bottom, left, right) and loop until traversal completes.  
   • **YouTube Solution (Java)**: [Spiral Matrix – Java Solution](https://www.youtube.com/watch?v=w3iZyl3fHmU)
2. [Number of Islands](https://leetcode.com/problems/number-of-islands/)  
   • **Problem**: You are given a 2D grid of '1' (land) and '0' (water). An island is formed by connecting adjacent lands horizontally or vertically. Return the total number of islands.  
   • **Objective**: Implement DFS or BFS to traverse each connected component and count islands efficiently.  
   • **YouTube Solution (Java)**: [Number of Islands – Java Solution](https://www.youtube.com/watch?v=U6-X_QOwPcs)

**Submission Checklist  
• Time and space complexity analysis for each solution.  
• Test cases demonstrating the correctness of your solutions.**