ಭಾರತೀಯ ಮಾಹಿತಿ ತಂತ್ರಜ್ಞಾನ ಸಂಸ್ಥೆ ರಾಯಚೂರು
भारतीय सूचना प्रौद्योगिकी संस्थान रायचूर
Indian Institute of Information Technology Raichur

LAB-7
Introduction to OOP

Submission Guidelines

- Ensure your system is in 'No Aeroplane Mode'.
- No Taskbar should be open.
- Create a new folder named `LAB-7`.
- Inside `LAB-7`, create two question files named in the following format:
  `[RollNumber]_[LabName]_[QuestionNumber]`
  (e.g., `12345_MatrixLab_Q1.cpp` and `12345_MatrixLab_Q2.cpp`)

Lab Timing and Submission

- Lab Time: **6 pm to 8PM**
- Submission Deadline:8:10PM (Submit on Classroom)
- No Extensions: Late submissions will not be accepted.
- Viva

Question 1:-

Design a stack that supports push, pop, top, and retrieving the minimum element in constant time.
Implement the **MinStack class:**

MinStack() :initializes the stack object.

void push(int val) :pushes the element val onto the stack.

void pop() :removes the element on the top of the stack.

int top(): gets the top element of the stack.

int getMin() :retrieves the minimum element in the stack.

You must implement a solution with O(1) time complexity for each function.

Example 1:

Input

["MinStack","push","push","push","getMin","pop","top","getMin"]

[[],[-2],[0],[-3],[],[],[],[]]

Output

[null,null,null,null,-3,null,0,-2]

Explanation:

MinStack minStack = new MinStack();

minStack.push(-2);

minStack.push(0);

minStack.push(-3);

minStack.getMin(); // return -3

minStack.pop();

minStack.top();   // return 0

minStack.getMin(); // return -2

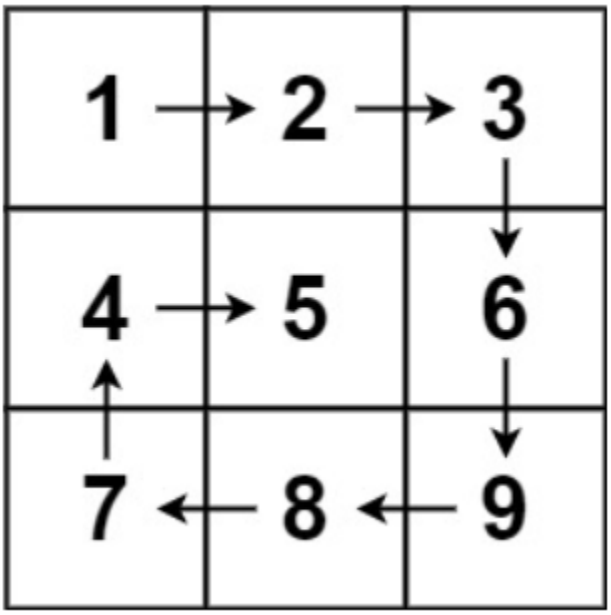
Constraints:

- -2^31 <= val <= 2^31 - 1
- Methods pop, top and getMin operations will always be called on non-empty stacks.
- At most 3 * 104 calls will be made to push, pop, top, and getMin.

Question 2:-

Given an m x n matrix, return all elements of the matrix in spiral order.

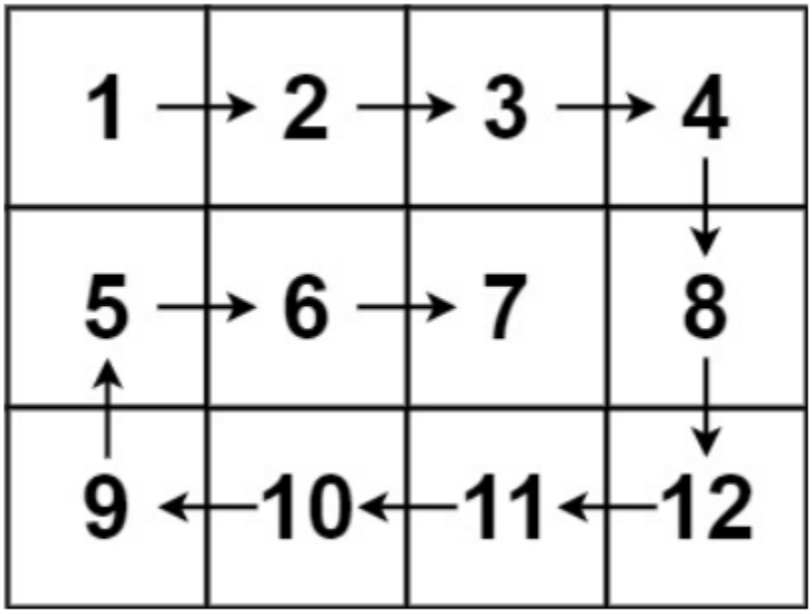
Example 1:



Input: matrix = [[1,2,3],[4,5,6],[7,8,9]]

Output: [1,2,3,6,9,8,7,4,5]

Example 2:



Input: matrix = [[1,2,3,4],[5,6,7,8],[9,10,11,12]]

Output: [1,2,3,4,8,12,11,10,9,5,6,7]

Constraints:

- m == matrix.length
- n == matrix[i].length
- 1 <= m, n <= 10
- -100 <= matrix[i][j] <= 100