1. Maximum Product of Two Elements in an Array.
   Given the array of integers nums, you will choose two different indices i and j of that array. Return the maximum value of (nums[i]-1)*(nums[j]-1).
   Example 1: Input: nums = [3,4,5,2] Output: 12
   Explanation: If you choose the indices i=1 and j=2 (indexed from 0), you will get the maximum value, that is, (nums[1]-1)*(nums[2]-1) = (4-1)*(5-1) = 3*4 = 12.
   Example 2: Input: nums = [1,5,4,5] Output: 16
   Explanation: Choosing the indices i=1 and j=3 (indexed from 0), you will get the maximum value of (5-1)*(5-1) = 16.
   Example 3: Input: nums = [3,7] Output: 12

2. Minimum Size Subarray Sum.
   Given an array of positive integers nums and a positive integer target, return the minimal length of a
   subarray
   whose sum is greater than or equal to target. If there is no such subarray, return 0 instead.

Example 1: Input: target = 7, nums = [2,3,1,2,4,3] Output: 2

Explanation: The subarray [4,3] has the minimal length under the problem constraint.

   Example 2: Input: target = 4, nums = [1,4,4] Output: 1
   Example 3: Input: target = 11, nums = [1,1,1,1,1,1,1] Output: 0

3. Minimum Window Substring.
   Given two strings s and t of lengths m and n respectively, return the minimum window substring
   of s such that every character in t (including duplicates) is included in the window. If there is no such substring, return the empty string "".

   The testcases will be generated such that the answer is unique.

   Example 1: Input: s = "ADOBECODEBANC", t = "ABC" Output: "BANC"
   Explanation: The minimum window substring "BANC" includes 'A', 'B', and 'C' from string t.

   Example 2: Input: s = "a", t = "a" Output: "a"
   Explanation: The entire string s is the minimum window.

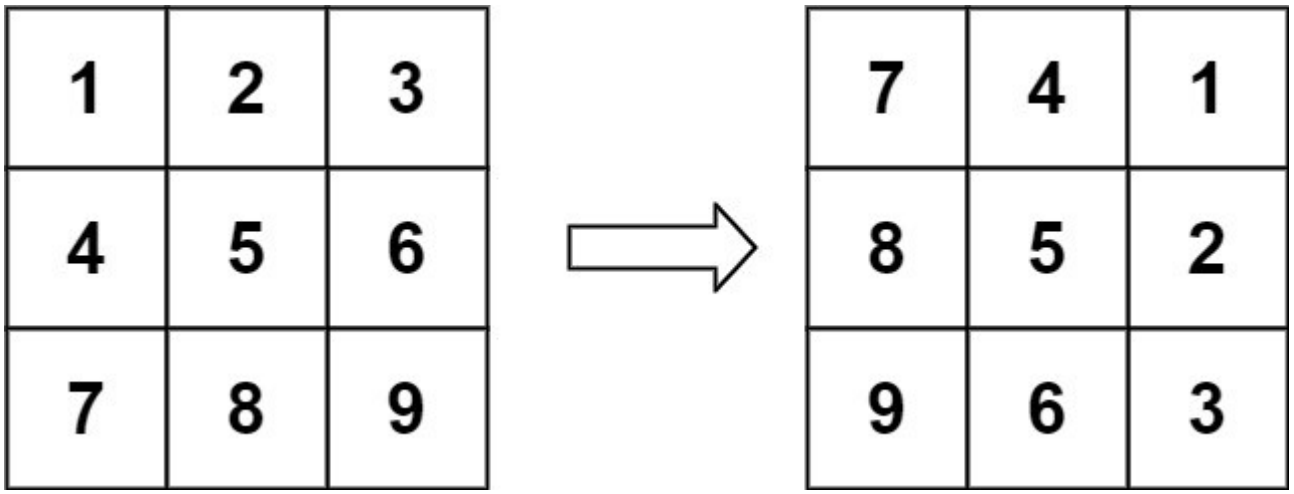
   Example 3: Input: s = "a", t = "aa" Output: ""
   Explanation: Both 'a's from t must be included in the window.
   Since the largest window of s only has one 'a', return empty string.

4. Rotate Image.

You are given an n x n 2D matrix representing an image, rotate the image by **90** degrees (clockwise).

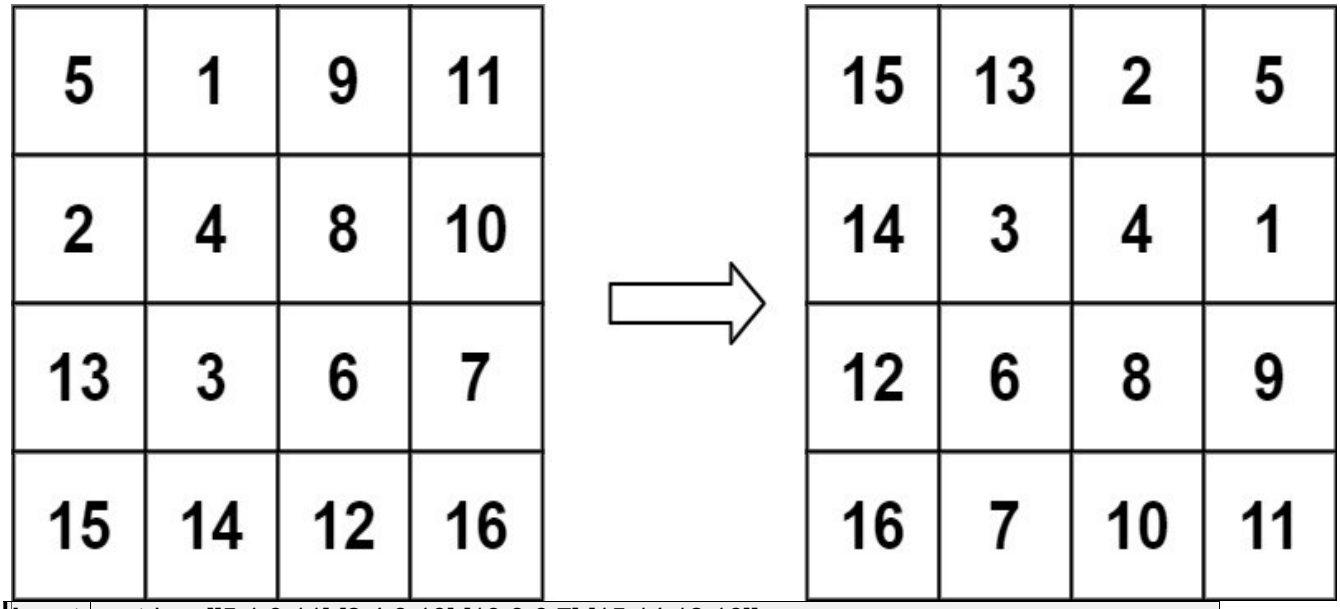
You have to rotate the image **in-place**, which means you have to modify the input 2D matrix directly. **DO NOT** allocate another 2D matrix and do the rotation.

Example 1:



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

Example 2:



Input: matrix = [[5,1,9,11],[2,4,8,10],[13,3,6,7],[15,14,12,16]]
Output: [[15,13,2,5],[14,3,4,1],[12,6,8,9],[16,7,10,11]]