1. Next grater element

    ip : 2, 1, 8, 7, 6, 5
    op : 2,5,1,6,7,8

    Ip : 5, 4 ,3 ,2,1
    op : 5,4,3,2,1

    ip: 1,2,3,4,5
    op : 1,2,3,5,4

    ip: 1, 2, 3, 4, 5, 1
    op: 1, 2, 3, 5, 1, 4

2. Print all diagonals in this way :

    ip: {{1,2,3},{4,5,6},{7,8,9}}
    op : {9,8,6,7,5,3,4,2,1}

3. Print SubArray window length which sum is max

4. Move all zeros to left side , with matin the order

    ip: {0,1,0,10,9,0,0,2}
    op: {0,0,0,0,1,10,9,2}

5. Print Product of given array except that index position value

Given an integer array nums, return *an array* answer *such that* answer[i] *is equal to the product of all the elements of* nums *except* nums[i].

The product of any prefix or suffix of nums is **guaranteed** to fit in a **32-bit** integer.

You must write an algorithm that runs in O(n) time and without using the division operation.

Example 1:
Input: nums = [1,2,3,4]
Output: [24,12,8,6]
Example 2:
Input: nums = [-1,1,0,-3,3]
Output: [0,0,9,0,0]