**Programming Problems for InfyTQ Round 2**

Q1. Given an array, only rotation operation is allowed on array. We can rotate the array as many times as we want. Return the maximum possible summation of i*arr[i].

**Examples :**
Input: arr[] = {1, 20, 2, 10}

Output: 72

We can get 72 by rotating array twice.

{2, 10, 1, 20}

20*3 + 1*2 + 10*1 + 2*0 = 72


Input: arr[] = {10, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Output: 330

We can get 330 by rotating array 9 times.

{1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

0*1 + 1*2 + 2*3 ... 9*10 = 330

Q2. An element in a sorted array can be found in O(log n) time via <u>binary search</u>. But suppose we rotate an ascending order sorted array at some pivot unknown to you beforehand. So for instance, 1 2 3 4 5 might become 3 4 5 1 2. Devise a way to find an element in the rotated array in O(log n) time.

Q3. An array contains both positive and negative numbers in random order. Rearrange the array elements so that positive and negative numbers are placed alternatively. Number of positive and negative numbers need not be equal. If there are more positive numbers they appear at the end of the array. If there are more negative numbers, they too appear in the end of the array.

For example, if the input array is [-1, 2, -3, 4, 5, 6, -7, 8, 9], then the output should be [9, -7, 8, -3, 5, -1, 2, 4, 6]

Q4. Given an array, for each element find the value of the nearest element to the right which is having a frequency greater than as that of the current element. If there does not exist an answer for a position, then make the value '-1'.

**Examples:**
**Input :** a[] = [1, 1, 2, 3, 4, 2, 1]
**Output :** [-1, -1, 1, 2, 2, 1, -1]
**Explanation:**
Given array a[] = [1, 1, 2, 3, 4, 2, 1]
Frequency of each element is: 3, 3, 2, 1, 1, 2, 3
Lets calls Next Greater Frequency element as NGF
1. For element a[0] = 1 which has a frequency = 3,
   As it has frequency of 3 and no other next element
   has frequency more than 3 so '-1'
2. For element a[1] = 1 it will be -1 same logic
   like a[0]
3. For element a[2] = 2 which has frequency = 2,
   NGF element is 1 at position = 6  with frequency
   of 3 > 2
4. For element a[3] = 3 which has frequency = 1,
   NGF element is 2 at position = 5 with frequency
   of 2 > 1
5. For element a[4] = 4 which has frequency = 1,
   NGF element is 2 at position = 5 with frequency
   of 2 > 1
6. For element a[5] = 2 which has frequency = 2,
   NGF element is 1 at position = 6 with frequency
   of 3 > 2
7. For element a[6] = 1 there is no element to its
   right, hence -1

**Input :** a[] = [1, 1, 1, 2, 2, 2, 2, 11, 3, 3]
**Output :** [2, 2, 2, -1, -1, -1, -1, 3, -1, -1]