Note: please follow the framework

1. Understand the question
2. Understand Input / Output
3. Prepare the solution in rough
4. Write clean code in book
5. Dry run and write output
6. Match the output
7. Write code in laptop and run, if output not matching debug and fix

Question 1:

Find an element in a sorted and rotated Array.

Expected time complexity : O(logN), Auxiliary space complexity: O(1)

Example 1:

Input: arr[] = {7, 10, 17, 22, 40, 1, 3, 5}, value = 3
Output: Present

Example 2:

Input: arr[] = {7, 10, 17, 22, 19, 40, 1, 3, 5}, value = 11
Output: Not Present

Question 2:

Write a program to print the next greatest elements in a given array. Elements for which no superior element exists, consider the next greatest element as -1.

Expected time complexity : O(N^2), Auxiliary space complexity: O(1)

Example 1:

Input: arr[] = {5,3,10,9,6,13}
Output: 10,10,13,13,13,-1
Explanation:
for 5 -> 10 is next greatest element

for 5 -> 10 is next greatest element
for 10 -> 13 is next greatest element
for 13 -> no next greatest element, so -1

Example 2:

Input: arr[] = {50,2,5,7,3,6}
Output: -1,5,7,-1,6,-1

Question 3:

Write a program to find the two repeating elements in a given array.

Expected time complexity : O(N^2), Auxiliary space complexity: O(1)

Example 1:

Input: arr[] = {2,7,4,7,8,3,4,8,9,9}
Output: 7,4

Example 2:

Input: arr[] = {7,33,2,6,10,33,17,2}
Output: 33,2

Question 4:

Write a program to find the majority element of an array.

Expected time complexity : O(N^2), Auxiliary space complexity: O(1)

Example 1:

Input: arr[] = {1,3,3,7,4,3,2,3,3,2,7,7}
Output: 3
Explanation: 3 is coming 5 times which is highest among all