CS212/CS514 - Assignment 2:

Q1: ***Convex Hull Computation:***

Problem statement:
You need to implement and analyze algorithm for computing the convex hull of a set of 2D points. The convex hull is a fundamental concept in computational geometry and is widely used in various applications. You need to find out the convex hull pertaining to the inputs and outputs defined as below in utmost O(n.log(n)) time complexity:

Input:
The input consists of the following components:
1. An integer, `n` (1 ≤ n ≤ 10^5), representing the number of 2D points.
2. `n` lines, each containing two floating-point numbers, `x` and `y`, separated by a space. These represent the coordinates of the 2D points. (-10^3 ≤ x, y ≤ 10^3)

Output:
Your program should produce the following output:
1. The list of points that make up the convex hull in counterclockwise order.
2. The total number of points on the convex hull.

Definitions:

- Convex Hull: It is a geometric shape that encloses a set of points in such a way that the shape is convex, meaning that any line segment connecting two points inside or on the boundary of the shape lies completely inside the shape itself. In simpler terms, the convex hull is the smallest convex polygon or polyhedron that contains all the given points.
-

Note:. Your implementations should handle all edge cases, including collinear/duplicate points.

Sample test cases:

Test Case 1:	Test Case 2:	Test Case 3:
Input	**Input**	**Input**
5 0 0 1 1 2 2 1 0 2 0	8 0 0 1 1 2 2 3 3 4 4 5 5 6 6 7 7	10 0 0 1 2 2 4 4 3 5 1 7 0 -1 -1 3 -2 2 -4 0 -3

Output 1:	**Output 2:**	**Output 3:**
Convex Hull: **0 0** **1 0** **2 0** **2 2** **0 0** **Total Points on Convex Hull: 4**	**Convex Hull:** **0 0** **7 7** **0 0** **Total Points on Convex Hull: 2**	**Convex Hull:** **-1 -1** **0 -3** **7 0** **2 4** **-1 -1** **Total Points on Convex Hull: 4**

You can use these test cases to evaluate the correctness of your convex hull computation algorithms.

Q2: ***Radix Sort:***

Problem Statement:

You are tasked with implementing the Radix Sort algorithm, which is a non-comparative sorting algorithm commonly used for integers and strings.

Input:

The input consists of the following components:

1. An integer, `n` (1 ≤ n ≤ 10^5), representing the number of integers to be sorted.
2. `n` integers, separated by spaces, representing the list of integers to be sorted. (-10^6 ≤ Integer values ≤ 10^6)

Output:

Your program should produce the following output:

1. The list of integers sorted in ascending order.

Definitions:

- Radix Sort:it is an integer (non-comparative) sorting algorithm that sorts integers or strings by processing individual digits or characters from the least significant to the most significant. It relies on the concept of "buckets" to distribute elements based on their digits or characters, repeatedly sorting them until the entire list is sorted.

Sample Test Cases:

Test Case 1:	**Test Case 2**:
- Input: 6 170 45 75 90 802 24 *- Expected Output:* Sorted List: 24 45 75 90 170 802	*- Input:* 10 -170 45 -75 90 802 24 -48 981 0 43 *- Expected Output:* Sorted List: -170 -75 -48 0 24 43 45 90 802 981

Q3) <u>Median of Medians</u>

Given an array income[] of size N which denotes the money earned by N individuals in a country. Government decides to exempt tax for people whose salary is less than Kth largest income.

a) So, find out the minimum income of the person who has to pay taxes.
b) One of the people wants to find his/her rank based on the income M so also find that out, given the person's income and it is part of the income[] array.

Time complexity for the given problem should be linear (O(n)).

Constraints
1≤ N ≤ 1000
1≤ K ≤ N
1≤ M ≤ 1e9
1≤ income[i] ≤ 1e9

Input
First line contains three space separated integers N, K and M denoting the size of the array ,the parameter K in the statement and parameter M in (b). Next line contains N space separated integers where i^{th} number denotes the i^{th} element of input array ,income[i].

Output
Output for part (a) should be a single integer denoting the found K th largest element of the array.
Output for part (b) should be a single integer denoting the Rank of the person with the given income.
Note

1. The input array arr may or may not be sorted & can have duplicates too. You have to find the Kth largest element in sorted order, not the distinct Kth largest element.
2. Also since elements can have the same value so in case of multiple ranks for a given value return the smallest rank.

Example
Sample 1:
6 3 7
1 7 7 4 4 4
Output:
4
5
Explanation:
Sorted array : 1 4 4 4 7 7
So the 3rd largest number is 4. So the minimum salary to pay tax is 3 . And 7 rank can be 5 or 6 so the smallest value is 5.

Sample 2:

10 7 2
2 8 95 864376 43542 654653 37833 3659 1 56

Output:

56
9

Submission Guidelines:

1. Submit your source code files.
2. Provide a brief report discussing the time complexity of the algorithms used, intuition/source used for understanding the same; any other knowledge you gain from the implementation.
3. Submit zip file with naming format: “CS212_A2_<EnrollmentNo>”

Note: Plagiarism will not be tolerated. Ensure that your code and report are your original work.

Good luck!