

Problem C. Orden de altura

Time limit 1000 ms
Mem limit 1048576 kB
OS Linux

José siempre hace que su clase se forme en orden de altura (el más bajo al frente de la línea). Cada septiembre llega una nueva clase de exactamente 20 estudiantes de tercero básico, todos con alturas distintas.

Durante los primeros días, lleva mucho tiempo poner a los niños en orden de altura, ya que nadie sabe dónde deben estar en la línea. No hace falta decir que hay mucho movimiento y reajuste.

Este año, José decidió probar un nuevo método para minimizar este caos en el ordenamiento. Se seleccionaría a un estudiante para que sea la primera persona en la línea. Luego, se seleccionaría otro estudiante que buscaría a la primera persona en la línea que sea más alta que él, y se colocaría frente a esa persona, lo que haría que todos los estudiantes detrás de él retrocedieran para darle espacio. Si no hay ningún estudiante más alto, entonces él se colocaría al final de la línea.

Este proceso continúa, un estudiante a la vez, hasta que todos los estudiantes estén en línea, momento en el cual los estudiantes estarán ordenados por altura.

Para este problema, escribirás un programa que calcule el número total de pasos dados hacia atrás durante el proceso de ordenamiento para una clase dada de estudiantes.

Input

La primera línea de entrada contiene un solo número entero T ($1 \leq T \leq 1000$), que es el número de casos de prueba que le siguen. Cada caso de prueba debe procesarse de forma independiente.

Cada caso de prueba consta de una sola línea de entrada. Contiene el número del caso de prueba (N), seguido de 20 enteros no negativos y únicos separados por un solo espacio. Los 20 enteros representan la altura (en milímetros) de cada estudiante en la clase.

Output

Para cada caso de prueba, hay una línea de salida.

La línea de salida única consta del número del caso de prueba, N , seguido de un espacio y luego del número total de pasos dados hacia atrás.

Sample 1

Input																					Output
4																					1 0
1	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	2 190
2	919	918	917	916	915	914	913	912	911	910	909	908	907	906	905	904	903	902	901	900	3 19
3	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	900	4 171
4	918	917	916	915	914	913	912	911	910	909	908	907	906	905	904	903	902	901	900	919	