Problem A. Height Ordering

Time limit	1000 ms
Mem limit	1048576 kB
OS	Linux

Mrs. Chambers always has her class line up in height order (shortest at the front of the line). Every September a new class of exactly 20 3rd graders arrive, all of different height. For the first few days it takes a long time to get the kids in height order, since no one knows where they should be in the line. Needless to say, there is quite a bit of jockeying around. This year Mrs. Chambers decided to try a new method to minimize this ordering chaos. One student would be selected to be the first person in line. Then, another student is selected and would find the *first* person in the line that is taller than him, and stand in front of that person, thereby causing all the students behind him to step back to make room. If there is no student that is taller, then he would stand at the end of the line. This process continues, one student at-a-time, until all the students are in line, at which point the students will be lined up in height order.

For this problem, you will write a program that calculates the total number of steps taken back during the ordering process for a given class of students.

Input

The first line of input contains a single integer P, $(1 \leq P \leq 1000)$ which is the number of data sets that follow. Each data set should be processed identically and independently.

Each data set consists of a single line of input. It contains the data set number, K, followed by 20 non-negative unique integers separated by a single space. The 20 integers represent the height (in millimeters) of each student in the class.

Output

For each data set there is one line of output. The single output line consists of the data set number, K, followed by a single space followed by total number of steps taken back.

Sample 1

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