

Problem A

DA-Sort

Problem ID: dasort
CPU Time limit: 1 second
Memory limit: 1024 MB

Source: 2016 Greater N
Region ACM Regional C
License: For educational

You recently learned a new way to sort an array of numbers in your algorithms course. The algorithm sorts an array of numbers by repeatedly performing the *Delete-and-Append* operation. The *Delete-and-Append* operation consists of three steps:

1. Choose an element from the array.
2. Delete the chosen element from the array.
3. Append the chosen element to the end of the array.

Being a curious student, you wonder what is the minimum number of *Delete-and-Append* operations required to sort a given array.

Input

The first line of input contains a single decimal integer P , ($1 \leq P \leq 100$), which is the number of data sets that follow. Each data set should be processed identically and independently.

Each data set consists of two or more lines of input. The first line contains the data set number, K , followed by a single space, followed by an integer N , ($1 \leq N \leq 1000$), which is the length of the array to sort. The remaining lines in the dataset contains N positive integers that comprise the array to be sorted, 10 values per line, except for the last line which may have less than 10 values. All the array elements are no larger than 10^9 . The same value may appear more than once in the array to be sorted.

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 an integer which is the minimum number of *Delete-and-Append* operations required to sort the array.

Sample Input 1

```
3
1 3
1 3 2
2 6
1 5 2 4 3 6
3 23
67890 56312 999999999 12345 23456 38927 45632 100345 98765 23456
87654 43278 23456 117654 321899 25432 54326 217435 26845 31782
33456 41234 56213
```

Sample Output 1

```
1 1
2 3
3 15
```