Problem C : Entropy

inputfile: input.C
outputfile: output.C

Description

After her brilliant discovery of worm holes exactly one year ago, the physicist returns to her daily work, which consists mainly of doing experiments and analysing the measurement data. Because of the low funding of her research group she has to use simple devices like a cheap Entr-o-Meter™ to measure the entropy of the universe. Despite the second law of thermodynamics, which states that entropy must always be increasing, her measurement data does not exhibit that pattern. She suspects that the cheap Entr-o-Meter has a loose connection and that only some of the values in the measurement data actually are entropy values, and the rest is just noise. She needs your help to extract the entropy data from her measurements. Buying a more expensive Analyz-o-Matic® is not an option.

Problem

The measurements consist of sequences of seemingly random numbers. The first step in the analysis, for which your help is needed, is to determine the number of valid data elements in each of these sequences. The physicist's hypothesis is that, if you take a selection of the measurement values, the longest strictly increasing selection is the 'real' entropy data. For example, if the measurements are `6 4 5 5 4 10`, the hypothesis says that the entropy values are `4 5 10`. Of course, this is still a hypothesis. To test the hypothesis, the physicist wants you to write a program that determines for each of the measurement sequences the *length* of such a selection.

Input

The input file starts with a line containing the number of measurement sequences to be analysed.

Each measurement sequence starts with a line containing the number of measurement values n ($1 \le n \le 1000$), followed by a line containing the actual measurement values, integers between -1000000 and 1000000, separated by single spaces.

Output

The output file should contain one line for each measurement sequence, containing the length of the longest strictly increasing selection of measurement values from that sequence.

Example

Sample input

```
4
6
6 4 5 5 4 10
3
0 -1000000 0
6
1 1 2 2 3 3
6
3 3 2 2 1 1
```

Correct output for the sample input

```
3
2
3
1
```