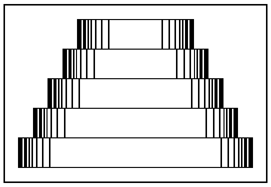
Problem 16: You Take the Cake



Source filename:	`cake.(cpp|java)`
Input filename:	`cake.in`
Output filename:	`cake.out`

As an employee of MCI (Mammoth Cakes Incorporated), it is your job to create extremely large layered birthday cakes. A layered birthday cake is made by taking small circular cakes layers and stacking them on top of each other. To perform your job, you stand in front of a big conveyor belt while layers of varying sizes pass in front of you. When you see one you like, you may take it off the conveyor belt and add it to your cake. You may add as many layers to your cake as you would like, as long as you follow these rules:

- Once a layer is added to your cake it cannot be moved. (It messes up the icing.) Thus, layers can only be added to the top of your cake.

- Each layer passes in front of you only once. You may take it or leave it. If you take it, you must add it to the top of your cake. If you leave it, it will move on down the conveyor belt, never to return.

- Each layer in your cake must be at least as small as the layer below. You cannot place a larger layer on top of a smaller one.

You will be told in advance the diameters (in inches) of the layers coming down the conveyor belt. Your job is to create the tallest cake possible using those layers.

For example, suppose the following list represents the diameters of the layers coming down the conveyor belt: `8 16 12 6 6 10 5`
Suppose you take the first layer (with a diameter of 8") for your cake. That means you may not take the second layer (since you already have a layer of size 8", and 16" > 8"). Similarly, you could not take the third layer, but you could take the fourth layer (since 6" < 8"). Following that, you could also take the fifth layer (the rule is simply that the layer on top cannot be larger; it can be the same size). Proceeding in this fashion we can create a cake with a height of 4 layers: `8 6 6 5`

However, if we had let the first layer go on by and started with the second layer, we could create a cake with a height of 5: `16 12 6 6 5`

The input for this problem is a list of the diameters of cake layers as they come down the conveyor belt. Your job is to determine the maximum height of all the cakes that could be built from those layers in that particular order. (Being a truly expert cake maker, you are not hampered by such little details as the laws of physics. As such, there is no limit on the size of cake you can create. If you get a thousand layers you can use, you can make a cake with a thousand layers.)

Write a program named **`cake.cpp`** that reads input from a file named **`cake.in`** and writes output to a file named **`cake.out`**. All output to the screen will be ignored.

Input File (`cake.in`)

Your program will process multiple input sets, one per line. Each line will begin with an integer N, followed by N positive integers representing the sizes of the cake layers in the order that they will be arriving on the conveyor belt. N will always be a non-negative integer, $0 \leq N \leq 100{,}000$. Each layer will have a diameter between 1 and 100,000, inclusive. A line where $N = 0$ marks the end of the input.

Be careful! Using an inefficient algorithm may result in a program that will exceed the time limit for some test cases.

Output File (`cake.out`)

You will output one line per data set (except for the one with $N = 0$) that simply contains the height (in layers) of the tallest cake you could possibly make from the layers in that data set.

Sample Input

```
7 8 16 12 6 6 10 5
10 45 25 40 38 20 10 32 25 18 30
10 10 9 8 7 6 5 4 3 2 1
0
```

Sample Output

```
5
6
10
```