

The Mountain Logo's Peak

We say an array of integers has a *mountain* shape if it is composed of a sequence of integer numbers that is sorted in strictly ascending order, followed by another sequence of integer numbers sorted in strictly descending order (any of them might be empty) and the maximum element appears only once in the array. We call *peak of the mountain* to the position that contains the maximum element of a *mountain array*.

You must:

- **Code a recursive function** that returns the position of the peak of a given mountain array in $O(\log n)$ (logarithmic time with respect to the size of the array).
- **Specify** the former function.
- Calculate its **complexity** using the *iterative method* seen in class.

Input

Each test case is described in two lines. The first one contains the length of the array, which will always be $\leq 10^6$. The second line contains the elements of a mountain array, starting in position 0. The elements will always be positive.

The input ends with an array of length 0 which must not be processed.

Output

For each test case the program must write the *position of the peak* of the mountain in a separate line. Remember the positions of an array start in 0.

Sample Input

```
5
1 2 3 2 1
5
1 4 3 2 0
8
10 12 13 15 20 22 23 22
3
3 2 1
5
1 2 3 4 5
0
```

Sample Output

```
2
1
6
0
4
```

Note

This exercise must be understood in the context of the *Data Structures and Algorithms* course, FDI-UCM 2017/2018 (prof. Gonzalo Méndez). Therefore, only those solutions which use the concepts explained in this course are valid.