You are given an array of 'N' integers denoting the heights of the mountains. You need to find the length of the longest subarray which has the shape of a mountain.

A mountain subarray is defined as a subarray which consists of elements that are initially in ascending order until a peak element is reached and beyond the peak element all other elements of the subarray are in decreasing order.

Example:

```
If the given array is: [1 3 1 4]. The longest mountain subarray would be 3. This
is because the longest mountain is  [1 3 1] having length 3.
```

Input Format:

```
The first line of input contains a single integer 'T', representing the number of
test cases or queries to be run.

Then the 'T' test cases follow.

The first line of each test case contains a single integer 'N' representing the
length of the array.

The second line of each test case contains 'N' space-separated integers denoting
the elements of the given array.
```

Output Format:

```
For each test case, print the length of the longest subarray which has the shape
of a mountain in a seperate line.
```

Constraints:

```
1 ≤ T ≤ 10
1 ≤ N ≤ 10^5
1 ≤ Ai ≤ 10^9

Time Limit : 1 sec
```

Note:

```
You are not required to print the expected output, it has already been taken care
of. Just implement the function.
```

Sample Input 1:

```
3
4
1 3 1 4
6
1 3 1 4 3 1
3
3 1 3
```

Sample Output 1:

```
3
4
0
```

Explanation Of Input 1:

```
The first test case is already explained in the problem statement.
The second test case, the given array is: [1 3 1 4 3 1] and the longest mountain
would be of length: 4 i.e. 1 4 3 1.
The third test case, the given array is: [3 1 3] and the longest mountain would be
of length: 0 since there is no increasing, peak and decreasing subarray.
```

Sample Input 2:

```
3
4
4 5 1 3
5
4 5 6 7 8
4
9 3 5 4
```

Sample Output 2

```
3
0
3
```