

Meow-tain View (Easy Version)

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

This is the easy version of the problem. The only difference is that in this version $N \leq 10^3$.

There are N mountains in UM Moon, arranged in a row labelled from 1 to N . Each mountain has a distinct height, H_i , from 1 to N .

Red Mooncat would like to know the number of mountains that he can view at each mountain's peak. V_i represents the number of mountains that he can view from the i^{th} mountain. **At i^{th} mountain, Red Mooncat could only view j^{th} mountain if and only if there is no greater height of mountain than j^{th} mountain exists between i^{th} mountain and j^{th} mountain exclusively.**

Red Mooncat can always view the i^{th} mountain itself when he is viewing from the i^{th} mountain.



Image source: <https://mooncatrescue.com/>

Input

The first line contains a single integer, N ($1 \leq N \leq 10^3$) — the number of mountains in UM Moon.

The second line contains N integers, $H_1, H_2, H_3, \dots, H_N$ ($1 \leq H_i \leq N$), where H_i denotes the height of i^{th} mountain.

Output

Output N integers, $V_1, V_2, V_3, \dots, V_N$, where V_i denotes the number of mountains that Red Mooncat can view at i^{th} mountain's peak.

Example

standard input	standard output
5 3 2 4 1 5	4 4 5 3 3

Note

View from the 1st mountain & 2nd mountain: Red Mooncat can all the mountains except the 4th mountain (blocked by the 3rd mountain).

View from the 3rd mountain: Red Mooncat can view all the mountains.

View from the 4th mountain & 5th mountain: Red Mooncat can view the 3rd mountain, 4th mountain, and 5th mountain because the first 2 mountains were blocked by the 3rd mountain.

Note that Red Mooncat could view the 5th mountain from the 2nd mountain but could not view the 2nd mountain from the 5th mountain because the 3rd mountain has blocked it.