

CMPUT 275 - Tangible Computing

Morning Problem: Climb

Description

Carla is going on a bike tour in the mountains and she wants to prepare. She has collected information about the elevation along the route measured at equidistant points. However, she is a little worried about the climbs. She decides that it would be nice to have information about how long the road is going to go up (or *climb*) before it either levels or starts to go down. She would like to know this information for each given point along the road. Help Carla by writing code, which, given the elevations at waypoints along the road, prints for each waypoint for how many more waypoints the road will be going up.

Input

Each input will contain two lines. On the first line, there will be a single integer $1 \leq n \leq 1000$, the number of waypoints on the route. On the next line, there will be n space separated integers specifying the elevations at successive waypoints. Each of these integers will fit in a 32-bit signed integer: they may be negative (i.e. a valley below sea level).

Output

For each waypoint, print the number of waypoints before the road levels or starts to go down. For the last point, the output should be zero. The numbers in the output should be separated by a single space (but there should be no trailing space in the output).

Sample Input 1

```
10
300 290 285 290 295 300 300 300 299 301
```

Sample Output 1

```
0 0 3 2 1 0 0 0 1 0
```

Explanation: For the first two waypoints, the road is going down. From the third waypoint (at elevation 285), the road is going up until the 6'th waypoint (which is at elevation 300). Specifically, it takes 3 more waypoints before the climb stops. Then the road is level again, going down and just before the end there is one more climb of length one (301 is the final elevation: no climbing from here, hence the output is zero).

Sample Input 2

```
4
1000 1020 1040 1001
```

Sample Output 2

2 1 0 0

Explanation: From 1000 to 1040, the elevation is increasing, so the output for the first position is 2 and the second is 1. The next two are 0 because the path decreases until the end.