Problem B. Ladder Master

Time limit 2000 ms
Mem limit 65536 kB

You need to find the minimum strength factor (k) required to reach the top rung of a ladder. The ladder has rungs at various heights from the ground. You can jump from the current rung (or the ground) to the next rung only, and you can't skip rungs. If you jump exactly (k) feet, (k) is decremented by 1. If you jump less than (k) feet, (k) remains the same.

For example, let the height of the rungs from the ground be 1, 3, respectively and **k** be 2. Now some steps are:

1. Jumped 1 foot from the ground to the 1^{st} rung (ground to 1). Since you jumped less than **k** feet, **k** remains 2.
2. Jumped 2 feet for the next rung (1 to 3). So, **k** becomes 1.

Input

Input starts with an integer **T (≤ 10)**, denoting the number of test cases.

Each case starts with a line containing an integer **n** $(1 \le n \le 10^5)$ denoting the number of rungs in the ladder.

The next line contains **n** space separated integers, $r_1, r_2 \ldots, r_n$ $(1 \le r_1 < r_2 < \ldots < r_n \le 10^9)$ denoting the heights of the rungs from the ground.

Output

For each case, print the case number and the minimum possible value of **k** as described above.

Sample

Input	Output
2 5 1 6 7 11 13 4 3 9 10 14	Case 1: 5 Case 2: 6

Note

Dataset is huge, use faster I/O methods.