Problem A. Conga line

Input:	`standard input`
Output:	`standard output`
Time limit:	2 seconds

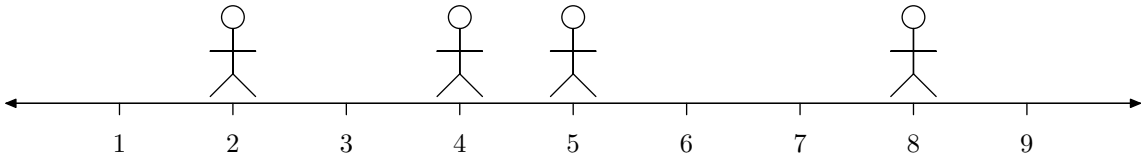
Conga is a traditional dance in which people make a line, grab each other by the waist and start dancing around.

You are at a party and your favorite Conga song starts playing. Since you want to make the most of it, you'd like to organize everybody and start dancing as soon as possible.

The dance floor is modeled as an infinite straight line with people standing on positive integer coordinates. There is at most one person at each point. Every second, a person can move one unit to the left or one unit to the right, as long as no one else is standing there. However, since it's a crazy party and people are already drunk, at most one person can move every second (in other words, no two people can move simultaneously).

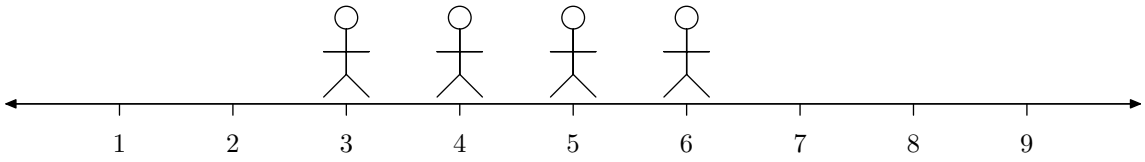
Nobody will start dancing until everybody is organized in a perfect line. You want to find the minimum amount of time it takes to start dancing, i.e. the time it takes to make people stand in such a way that there are no empty spaces between them.

For example, imagine there are 4 people at the party, standing at positions 2, 4, 5 and 8:



In this case, it takes at least 3 seconds to form the Conga line:

- On second 1, the person standing at position 2 moves to position 3.
- On second 2, the person standing at position 8 moves to position 7.
- On second 3, the person standing at position 7 moves to position 6.
- After three seconds, people are standing on positions 3, 4, 5, and 6 and they can start dancing!



Input

The input contains several test cases.

The first line of each case contains a single integer number n, the number of people in the party ($1 \leq n \leq 10^6$). The next line contains n distinct integers x_i separated by single spaces sorted in ascending order — the coordinates where people are initially standing ($1 \leq x_i \leq 10^9$).

The last line of the input contains a single 0 and should not be processed.

Output

For each test case, output one integer number on a single line — the minimum time it takes to start dancing.

Sample input and output

standard input	standard output
4 2 4 5 8 1 10 4 20 24 25 26 2 1 2 2 1 1000000000 0	3 0 3 0 999999998

Problem B. Secret word

Input:	`standard input`
Output:	`standard output`
Time limit:	2 seconds

Alicia and Roberto like to play games. Today, Roberto is trying to guess a secret word that Alicia chose. Alicia wrote a long string S in a piece of paper and gave Roberto the following clues:

- The secret word is a non-empty substring of S (possibly equal to S)
- S starts with the secret word reversed

Roberto knows Alice very well, and he's sure that if there are several possible secret words that satisfy the clues above, Alice must have chosen the longest one.

Can you help him guess the secret word?

Input

The first line of the input file contains a single integer number $T \leq 100$, the number of test cases.

T lines follow, each with a single string S. S will only contain lowercase English letters. The length of S will not exceed one million characters.

Output

For each test case, output the secret word in one line.

Sample input and output

standard input	standard output
6 unicamp brazil abcdba neversayeven neveroddoreven listentothesilence	u b ba even neveroddoreven sil

Explanation of the sample cases

- `unicamp`: if you take `u` and reverse it you get `u`. `unicamp` starts with `u`, so this satisfies the two clues above. Furthermore, `u` is the longest word that satisfies the two clues so it must be the secret word.
- `abcdba`: if you take `ba` and reverse it you get `ab`. `abcdba` starts with `ab` and there's no longer substring that satisfies the two clues.
- `neversayeven`: if you take `even` and reverse it you get `neve`. `neversayeven` starts with `neve` and there's no longer substring that satisfies the two clues.
- `neveroddoreven`: this case is a palindrome so if we reverse it we get the same string. `neveroddoreven` starts with `neveroddoreven` (since they are the same) and obviously there's no longer substring that satisfies that, so this is the secret word.
- `listentothesilence`: Notice the secret word might be somewhere in the middle of the big word.