Problem A. Conga line

Input:	`standard input`
Output:	`standard output`
Time limit:	3 seconds

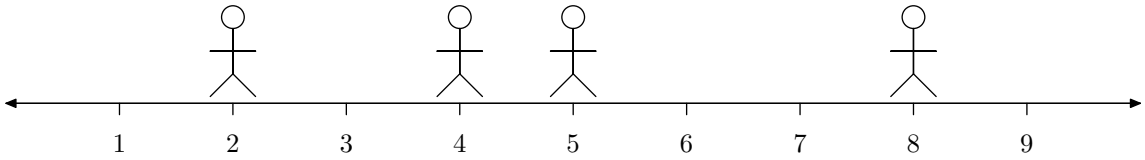
Conga is a traditional dance in which people make a line, grab each other by the waist and start dancing around.

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).

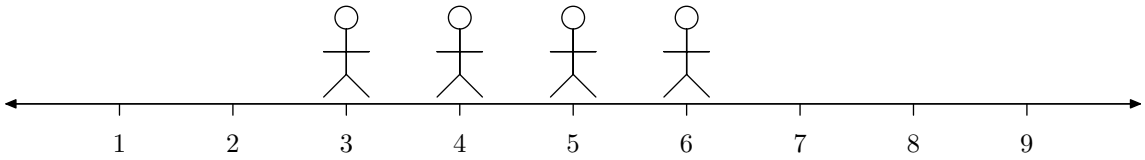
Nobody will start dancing until everybody is organized in a perfect line. You want to find the minimum amount of time it takes to start dancing, i.e. the time it takes to make people stand in such a way that there are no empty spaces between them.

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:	3 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 $T \leq 150$, 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.

Problem C. Stones

Input:	`standard input`
Output:	`standard output`
Time limit:	3 seconds

Alicia and Roberto like to play games. Today, they are playing a game where there's a pile of stones on the table. The two players alternate turns removing some stones from the pile. On the first turn, a player can remove any number of stones but he or she must leave at least one stone on the table. On following turns, a player can remove at most twice the number of stones that the other player removed on the previous turn. Each player must always remove at least one stone.

The player that takes the last stone wins.

For example, let's imagine there's a pile of 8 stones on the table, and let's imagine Alicia starts.

- On the first turn, Alicia must remove a number of stones between 1 and 7 (according to the rules above, on the first turn she can remove any number of stones as long as there remains at least 1). Let's say Alicia takes 2 stones, leaving 6 on the table.
- On the second turn, Roberto must remove at least 1 stone and at most 4 stones (because 4 is twice the number of stones that Alicia removed in the previous turn). Let's say he takes 3, leaving 3 stones on the table.
- On the third turn, Alicia must remove at least 1 and at most 6 stones (6 is twice the number of stones Roberto took in the previous turn). Since there remain only 3 stones on the table, she simply takes the 3 stones and wins the game!

However, Roberto could have played better. If he had taken only 1 stone on the second turn, he would have left 5 on the table with Alicia having to take between 1 and 2 stones. If Alicia took 2, Roberto could win immediately by taking the 3 stones that would remain. So Alicia's only choice is to take 1 stone, leaving 4 stones and Roberto having to take between 1 and 2 stones. Roberto would then take 1 stone, leaving 3 stones and Alicia having to take between 1 and 2 stones. Roberto could then win after Alicia's move, regardless of whether she takes 1 or 2 stones. In fact, it turns out that Roberto always has a winning strategy if there are 8 stones and Alicia plays first, even if Alicia plays in the best possible way!

Your task is to determine who wins the game if both players play optimally, assuming Alicia always plays first.

Input

The input contains several test cases.

Each test case contains a single integer n ($2 \leq n \leq 1000$), the number of stones that are initially on the table.

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

Output

For each test case, output `Alicia` or `Roberto` on a single line, depending on who wins if both players play optimally and Alicia plays on the first turn.

Sample input and output

standard input	standard output
8 2 4 986 987 0	Roberto Roberto Alicia Alicia Roberto