B. Shrinking Array

time limit per test: 2 seconds
memory limit per test: 256 megabytes

Let's call an array b *beautiful* if it consists of at least two elements and there exists a position i such that $|b_i - b_{i+1}| \leq 1$ (where $|x|$ is the absolute value of x).

You are given an array a, and as long as it consists of at least two elements, you can perform the following operation:

1. Choose two adjacent positions i and $i+1$ in the array a.
2. Choose an integer x such that $\min(a_i, a_{i+1}) \leq x \leq \max(a_i, a_{i+1})$.
3. Remove the numbers a_i and a_{i+1} from the array, and insert the number x in their place. Obviously, the size of the array will decrease by 1.

Calculate the minimum number of operations required to make the array beautiful, or report that it is impossible.

Input

The first line contains one integer t ($1 \leq t \leq 200$) — the number of test cases.

The first line of each test case contains one integer n ($2 \leq n \leq 1000$) — the size of the array a.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^6$) — the array a itself.

Output

For each test case, output one integer — the minimum number of operations needed to make the array a beautiful, or -1 if it is impossible to make it beautiful.

Example

input

```
4
4
1 3 3 7
2
6 9
4
3 1 3 7
4
1 3 5 2
```

output

```
0
-1
1
1
```

Note

In the first test case, the given array is already beautiful, as $|a_2 - a_3| = |3 - 3| = 0$.

In the second test case, it is impossible to make the array beautiful, as applying the operation would reduce its size to less than two.

In the third test case, you can, for example, choose a_1 and a_2 and replace them with the number 2. The resulting array $[2, 3, 7]$ is beautiful.

In the fourth test case, you can, for example, choose a_2 and a_3 and replace them with the number 3. The resulting array $[1, 3, 2]$ is beautiful.