Problem C. Subarray

Time limit 2000 ms
Mem limit 262144 kB

Let's call an array t *dominated* by value v in the next situation.

At first, array t should have at least 2 elements. Now, let's calculate number of occurrences of each number num in t and define it as $occ(num)$. Then t is dominated (by v) if (and only if) $occ(v) > occ(v')$ for any other number v'. For example, arrays $[1, 2, 3, 4, 5, 2]$, $[11, 11]$ and $[3, 2, 3, 2, 3]$ are dominated (by 2, 11 and 3 respectevitely) but arrays $[3]$, $[1, 2]$ and $[3, 3, 2, 2, 1]$ are not.

Small remark: since any array can be dominated only by one number, we can not specify this number and just say that array is either dominated or not.

You are given array $a_1, a_2, \ldots, a_n$. Calculate its shortest dominated subarray or say that there are no such subarrays.

The subarray of a is a contiguous part of the array a, i. e. the array $a_i, a_{i+1}, \ldots, a_j$ for some $1 \le i \le j \le n$.

Input

The first line contains single integer T ($1 \le T \le 1000$) — the number of test cases. Each test case consists of two lines.

The first line contains single integer n ($1 \le n \le 2 \cdot 10^5$) — the length of the array a.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$) — the corresponding values of the array a.

It's guaranteed that the total length of all arrays in one test doesn't exceed $2 \cdot 10^5$.

Output

Print T integers — one per test case. For each test case print the only integer — the length of the shortest dominated subarray, or -1 if there are no such subarrays.

Sample 1

Input	Output
4 1 1 6 1 2 3 4 5 1 9 4 1 2 4 5 4 3 2 1 4 3 3 3 3	-1 6 3 2

Note

In the first test case, there are no subarrays of length at least 2, so the answer is -1.

In the second test case, the whole array is dominated (by 1) and it's the only dominated subarray.

In the third test case, the subarray a_4, a_5, a_6 is the shortest dominated subarray.

In the fourth test case, all subarrays of length more than one are dominated.