

Majority sequence

A sequence of n integers a_0, a_1, \dots, a_{n-1} is said to have a majority element if there is an integer that occurs strictly more than $n/2$ times in the sequence.

Given a sequence of integers, you have to find the longest subsequence of the sequence that has a majority element. In the second part of the problem, you have to find the longest substring of the given sequence that has a majority element. A subsequence is obtained by deleting some elements of the sequence, keeping the order of the others the same. A substring is a consecutive sequence of elements a_i, a_{i+1}, \dots, a_j for some $0 \leq i \leq j < n$.

Input/Output: The first line of input will specify n the length of the input sequence, where $1 \leq n \leq 10^6$. The next line will contain the n numbers a_i , separated by a space, where $0 \leq a_i < n$ for $0 \leq i < n$. The first line of output should give the length of the longest subsequence with a majority element and the second line should give the length of the longest substring with a majority.

Please ensure that you print and flush out the output for the first part as soon as it is computed, before starting the second part.

Submission: Submit a single file named RollNo_1.cpp .