

Problem J. Judges Problem

Input file: judges.in
Output file: judges.out
Time limit: 3 seconds
Memory limit: 256 megabytes

Judges of Flatland University Contest have prepared an excellent problem set. But the morning before the contest they found out that computer virus has destroyed input files to one of the problems. Now they urgently need to restore the files. They decided to ask for your help. Given answer files for “Prefix Cover” problem you have to restore the input files.

The problem is stated as follows. Consider array $A = \langle a_1, a_2, \dots, a_n \rangle$ of integers. Array $B = \langle b_1, b_2, \dots, b_m \rangle$ is said to *cover* array A if it is possible for each i from 1 to n to find subarray of A equal to B containing a_i . For example, $A = \langle 1, 2, 1, 2, 1, 1, 2, 1 \rangle$ is covered by $B = \langle 1, 2, 1 \rangle$. Minimal m such that A can be covered by an array of length m is called *mincover* of A and denoted as $mc(A)$. In the example above $mc(\langle 1, 2, 1, 2, 1, 1, 2, 1 \rangle) = 3$.

Given A its prefix cover array $P = \langle p_1, p_2, \dots, p_n \rangle$ is an array of mincovers of A prefixes, that is $p_i = mc(\langle a_1, a_2, \dots, a_i \rangle)$. Prefix cover array for the array in the above example is $P = \langle 1, 2, 3, 2, 3, 6, 7, 3 \rangle$.

The judges wanted to ask teams to create prefix cover array for the given array. Now you have to solve the inverse problem: given an array P find some array A for which P is the prefix cover array.

Input

The input file contains multiple test cases.

The first line of each test case contains n ($1 \leq n \leq 5 \cdot 10^5$). The second line contains n integers p_i ($1 \leq p_i \leq i$).

The last test case is followed by a line containing zero, it must not be processed.

The sum of values of n in all test cases in one input file doesn't exceed $5 \cdot 10^5$.

Output

For each test case output “Yes” if there exists such array A that P is its prefix cover array. In such case the following line must contain n integers ranging from 1 to n — elements of A . If there is no such array, output “No”.

Examples

judges.in	judges.out
8	Yes
1 2 3 2 3 6 7 3	1 2 1 2 1 1 2 1
5	Yes
1 1 1 1 1	1 1 1 1 1
3	No
1 2 2	Yes
7	1 2 3 4 5 6 7
1 2 3 4 5 6 7	
0	