Problem F. Move to Front

Input file:	`mtf.in`
Output file:	`mtf.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

Move-to-Front is a method of transforming sequences of positive integer numbers, that is used in some compression algorithms, such as Burrows-Wheeler transform.

Initially all positive integer numbers are organized as an ordered list L in their natural order. Consider a sequence $a_1, a_2, \ldots, a_n$ of positive integer numbers. It is encoded as a sequence $b_1, b_2, \ldots, b_n$ in the following way. Let the part of the sequence from a_1 to a_{i-1} be encoded. The position of a_i in the current list L is considered. It is assigned to b_i, and a_i is moved to the beginning of the list L.

For example, the sequence $3, 3, 3, 2, 2, 2, 2, 2, 3, 1, 3, 3, 2$ is encoded as $3, 1, 1, 3, 1, 1, 1, 1, 2, 3, 2, 1, 3$.

You are given a sequence $a_1, a_2, \ldots, a_n$, you must encode it using Move-to-Front method, and output the resulting sequence $b_1, b_2, \ldots, b_n$.

Input

The first line of the input file contains integer number n ($1 \le n \le 100\,000$). The second line contains n integer numbers a_i, ranging from 1 to 10^9.

Output

Output n integer numbers — the sequence $b_1, b_2, \ldots, b_n$.

Example

mtf.in	mtf.out
13 3 3 3 2 2 2 2 2 3 1 3 3 2	3 1 1 3 1 1 1 1 2 3 2 1 3