Problem F. Finding Keys

Source file name:	Finding.c, Finding.cpp, Finding.java, Finding.py
Input:	`Standard`
Output:	`Standard`

Wolfgang Amadeus Mozart has too many keys! He has n keys of distinct lengths on his circular keychain. Unfortunately,Wolfgang can only judge whether a key fits into a door by its relative size compared to the keys surrounding it. Let the k-pattern of a key x be the sequence of relative key lengths of the k keys following key x in clockwise order on the keychain. For example, if keychain has keys of lengths 1, 5, 3, 4, 2 in clockwise order, then the 3-pattern of the key of length 3 can be expressed as the string “<>>”, since $3 < 4$, $4 > 2$, and $2 > 1$. Note that the last key of length 2 is followed by the first key of length 1.

Please help Wolfgang determine for each key the smallest k such that the k-pattern of the key is unique (no other key’s k-pattern is the same).

Input

The first line of input contains a single integer n ($2 \leq n \leq 2 \cdot 10^5$), the number of keys on Wolfgang’s circular keychain.

The next n lines each contain an integer between 1 and 10^9 representing the length of one key. The key lengths are given in their clockwise order on the keychain. It is guaranteed that all key lengths are unique.

Output

Output n lines, one integer per line. The i^{th} integer should be the smallest k such that the k-pattern of key i (in input order) is unique among all k-patterns. If there exists no such k, then the i^{th} integer should be -1.

Example

Input	Output
5 1 8 3 4 2	3 4 3 2 4
4 1 4 2 3	-1 -1 -1 -1