Problem A. Max or Min

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Kevin has n integers $a_1, a_2, \ldots, a_n$ arranged in a **circle**. That is, the numbers a_i and a_{i+1} $(1 \le i < n)$ are neighbors. The numbers a_1 and a_n are neighbors as well. Therefore, each number has exactly two neighbors.

In one minute, Kevin can set a_i to the minimum among three numbers: a_i and it's two neighbors. Alternatively, Kevin can set a_i to the maximum among the same numbers. For example, if $a_i = 5$ and a_i has two neighbors 3 and 2, and Kevin performs the minimum operation, a_i will be equal to 2. However, if he performs the maximum operation, a_i will remain 5.

For each x from 1 to m, find the minimum number of minutes to make all numbers equal x, or determine that it is impossible to do so.

Input

The first line contains two integers n and m $(3 \le n \le 2 \cdot 10^5, 1 \le m \le 2 \cdot 10^5)$ — the number of integers in the circle, and the number of integers you need to find answers for.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le m)$ — the integers in the circle.

Output

Print m integers. The i-th integer should be equal to the minimum number of minutes that are needed to make all numbers equal i or -1 if it's impossible.

Example

standard input	standard output
7 5 2 5 1 1 2 3 2	5 5 7 -1 6

Note

To make all numbers equal 2 Kevin needs at least 5 minutes. One of the possible sequence of operations:

1. Apply min operation to a_6, it will be equal to 2.
2. Apply max operation to a_4, it will be equal to 2.
3. Apply max operation to a_3, it will be equal to 5.
4. Apply min operation to a_2, it will be equal to 2.
5. Apply min operation to a_3, it will be equal to 2.