

Loss of Balance

Input File	Output File	Time Limit	Memory Limit
standard input	standard output	2 seconds	256 MiB

An array A is (x, y) -fair, if there is an index i and an index j such that:

- $i < j$, and
- $A_i = x$ and $A_j = y$.

An array A is K -balanced if and only if:

- All elements of the array are integers between 1 and K .
- Every integer from 1 to K appears *at least once* in A .
- A is (x, y) -fair and (y, x) -fair for all pairs of integers x and y (from 1 to K) where $x \neq y$.

For example:

- $[1, 2, 3, 4]$ is not 3-balanced, since the array contains a 4.
- $[1, 3]$ is not 3-balanced, since the array does not contain a 2.
- $[1, 2, 1, 3, 1]$ is not 3-balanced, since it is not $(3, 2)$ -fair.
- $[1, 2, 3, 2, 1]$ is 3-balanced.

Hugo has an array A containing N elements that is K -balanced. Unfortunately, he lost the array and would like you to help him recover it.

Fortunately, Hugo recalls a relative ordering of the elements of A . More precisely, he has an array B also of length N . For all i and j :

- if $B_i = B_j$, then $A_i = A_j$,
- if $B_i > B_j$, then $A_i \geq A_j$ (**large inequality**), and
- if $B_i < B_j$, then $A_i \leq A_j$ (**large inequality**).

For example:

- If $B = [1, 5, 2]$, then $A_1 \leq A_3 \leq A_2$.
- If $B = [6, 3, 7, 7, 3, 7]$, then $A_3 = A_4 = A_6$ and $A_2 = A_5$. Furthermore, $A_5 \leq A_1 \leq A_6$.

Please help Hugo recover A , or say that it is impossible!

Subtasks and Constraints

For all subtasks, you are guaranteed that:

- $2 \leq N \leq 200\,000$.
- $2 \leq K \leq N$.
- $1 \leq B_i \leq 200\,000$, for all i .

Additional constraints for each subtask are given below.

Subtask	Points	Additional constraints
1	10	$N \leq 10$ and $B_i \leq K$, for all i .
2	20	$B_i \leq K$, for all i .
3	30	$N \leq 3000$
4	10	$K \leq 5$
5	30	No further constraints apply.

Input

- The first line of input contains the two integers, N and K .
- The second line contains N integers. The i -th integer (starting from 1) is B_i .

Output

The output should contain N integers on a single line. The i -th of these integers should be A_i , and each integer **must** be between 1 and K .

The printed array A must be K -balanced **and** must match the relative order defined by B .

If there are many such correct array, you can output **any of them**. If no such array exists, print -1 instead.

Sample Input 1

```
5 3
5 300 900 300 6
```

Sample Output 1

```
1 2 3 2 1
```

Sample Input 2

```
10 5
1 2 3 4 5 6 7 8 9 10
```

Sample Output 2

```
-1
```

Sample Input 3

```
6 2
1 2 2 1 2 1
```

Sample Output 3

```
1 2 2 1 2 1
```

Explanation

In Sample Input 1, the **only** correct output is $[1, 2, 3, 2, 1]$ (it is 3-balanced **and** it matches the relative order of B).

Note that, for example, the three following arrays would be **incorrect** output:

- $[3, 1, 2, 3, 1]$ is 3-balanced, but does not match the relative order of B
- $[1, 2, 3, 2, 2]$ matches the relative order of B , but is not 3-balanced.
- $[1, 1, 2, 1, 1]$ matches the relative order of B , but is not 3-balanced.

In Sample Input 2, it is impossible to create an increasing 5-balanced array.

In Sample Input 3, the only correct output is $[1, 2, 2, 1, 2, 1]$. Note that Sample Input 3 satisfies the constraints of all subtasks.