E. Stack Sorting

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Let's suppose you have an array a, a stack s (initially empty) and an array b (also initially empty).

You may perform the following operations until both a and s are empty:

- Take the first element of a, push it into s and remove it from a (if a is not empty);
- Take the top element from s, append it to the end of array b and remove it from s (if s is not empty).

You can perform these operations in arbitrary order.

If there exists a way to perform the operations such that array b is sorted in non-descending order in the end, then array a is called *stack-sortable*.

For example, $[3, 1, 2]$ is *stack-sortable*, because b will be sorted if we perform the following operations:

1. Remove 3 from a and push it into s;
2. Remove 1 from a and push it into s;
3. Remove 1 from s and append it to the end of b;
4. Remove 2 from a and push it into s;
5. Remove 2 from s and append it to the end of b;
6. Remove 3 from s and append it to the end of b.

After all these operations $b = [1, 2, 3]$, so $[3, 1, 2]$ is *stack-sortable*. $[2, 3, 1]$ is not *stack-sortable*.

You are given k first elements of some permutation p of size n (recall that a permutation of size n is an array of size n where each integer from 1 to n occurs exactly once). You have to restore the remaining $n - k$ elements of this permutation so it is *stack-sortable*. If there are multiple answers, choose the answer such that p is lexicographically maximal (an array q is lexicographically greater than an array p iff there exists some integer k such that for every $i < k$ $q_i = p_i$, and $q_k > p_k$). **You may not swap or change any of first k elements of the permutation.**

Print the lexicographically maximal permutation p you can obtain.

If there exists no answer then output `-1`.

Input

The first line contains two integers n and k ($2 \le n \le 200000$, $1 \le k < n$) — the size of a desired permutation, and the number of elements you are given, respectively.

The second line contains k integers $p_1, p_2, ..., p_k$ ($1 \le p_i \le n$) — the first k elements of p. These integers are pairwise distinct.

Output

If it is possible to restore a *stack-sortable* permutation p of size n such that the first k elements of p are equal to elements given in the input, print lexicographically maximal such permutation.

Otherwise print `-1`.

Examples

input

```
5 3
3 2 1
```

output

```
3 2 1 5 4
```

input

```
5 3
2 3 1
```

output

```
-1
```

input

```
5 1
3
```

output

```
3 2 1 5 4
```

input

```
5 2
3 4
```

output

```
-1
```