Problem G – Gratitude Loop

The contest is over, but organizers and contestants are still thanking each other:

"Thank you!"
"No, thank you!"
"No, no, thank you!"
"No, no, no, thank you!"

This could go on forever! To save everyone some typing, they created a messaging app with just one button: Thanks!. After some effort, we were able to obtain a screenshot of the app installed on Paulo's device. We include it below for your convenience.



Of course, Paulo is not the only person using this wonderful app. There are N people using it, numbered from 1 to N. Initially, nobody has received any messages. Whenever person s presses Thanks!, that person sends exactly one message to a recipient r. The app chooses r as follows:

- If the sender s has already received any messages, then the recipient r is the person who most recently sent a message to s.
- If the sender s has not yet received any messages, then the recipient r is $s + 1$, unless $s = N$, in which case r is 1.

The message is delivered immediately, before the next button press, and the sender s becomes the person who most recently sent a message to the recipient r.

Receiving a message does not send an automatic reply. Sending a message does not delete or change any of the sender's received messages. A person can press the button repeatedly, even without receiving new messages between their button presses.

Given the sequence of button presses, determine how many messages each person receives. Count each message separately, including repeated messages from the same person.

Input

The first line contains two integers N ($2 \leq N \leq 1000$) and M ($1 \leq M \leq 1000$), indicating respectively the number of people and the number of button presses. Each person is identified by a distinct integer from 1 to N.

The second line contains M integers $A_1, A_2, \ldots, A_M$ ($1 \leq A_i \leq N$), where A_i is the person who makes the i-th button press. The presses are made in the given order.

Output

Output a single line with N integers $C_1, C_2, \ldots, C_N$, such that C_i is the number of messages received by person i during the M button presses.

Sample input 1	Sample output 1
4 4 1 2 3 2	2 1 0 1

Explanation of sample 1:

In this sample there are $N = 4$ people who make $M = 4$ button presses. The messages are sent from 1 to 2 (no messages received yet by 1), from 2 to 1 (last message received by 2 is from 1), from 3 to 4 (no messages received yet by 3), and from 2 to 1 (last message received by 2 is still from 1).

Sample input 2	Sample output 2
2 3 2 2 2	3 0

Explanation of sample 2:

Here $N = 2$ and $M = 3$. Person 2 never receives a message, so each of their three button presses sends a message to person 1.

Sample input 3	Sample output 3
3 4 1 2 3 1	2 1 1

Explanation of sample 3:

Now $N = 3$ and $M = 4$. The messages are sent from 1 to 2, from 2 to 1, from 3 to 1, and from 1 to 3.