

## 7589 Rearranging a Sequence

You are given an ordered sequence of integers,  $(1, 2, 3, \dots, n)$ . Then, a number of requests will be given. Each request specifies an integer in the sequence. You need to move the specified integer to the head of the sequence, leaving the order of the rest untouched. Your task is to find the order of the elements in the sequence after following all the requests successively.

### Input

The input file contains several test cases, each of them formatted as follows.

```
 $n$   $m$   
 $e_1$   
 $\vdots$   
 $e_m$ 
```

The integer  $n$  is the length of the sequence ( $1 \leq n \leq 200000$ ). The integer  $m$  is the number of requests ( $1 \leq m \leq 100000$ ). The following  $m$  lines are the requests, namely  $e_1, \dots, e_m$ , one per line. Each request  $e_i$  ( $1 \leq i \leq m$ ) is an integer between 1 and  $n$ , inclusive, designating the element to move. Note that, the integers designate the integers themselves to move, not their positions in the sequence.

### Output

For each test case, output the sequence after processing all the requests. Its elements are to be output, one per line, in the order in the sequence.

### Explanation for the Sample

In Sample Input 1, the initial sequence is  $(1, 2, 3, 4, 5)$ . The first request is to move the integer 4 to the head, that is, to change the sequence to  $(4, 1, 2, 3, 5)$ . The next request to move the integer 2 to the head makes the sequence  $(2, 4, 1, 3, 5)$ . Finally, 5 is moved to the head, resulting in  $(5, 2, 4, 1, 3)$ .

### Sample Input

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

**Sample Output**

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