01-D Betrayer's Badge

Time limit: 10s

In the Village of Eldoria, there is an ancient scroll that is believed to hold the secret locations of hidden treasures in the Kingdom. The king needs your help to decode the scroll. Instead of showing a map, is presents a list L of numbers. However, it is said that not all numbers are valid and one must cleanse the list to reveal the true message.

Legend has it that there was an infamous betrayer knight with a specific badge number, x, who attempted to sabotage the Kingdom a long time ago. To decode the scroll, the village elder instructs you with the following steps:

1. Read the list L from the ancient scroll containing the numerical representations.
2. Remove the entry at the position (index) equal to the betrayer knight's badge number x (positions on the list start from 0).
3. If the badge number x also appears as a number in the list L, remove all occurrences of it to ensure the betrayer's influence is completely wiped out.

Once cleansed, the list will guide the village to the hidden treasures. Can you help Eldoria decode the scroll?

Input

- The list of numbers L from the ancient scroll.
- x - Any integer, which is the badge number of the betrayer knight to be cleansed as described above.

Again, you can read the input using `list(map(int, input().split()))`.

Disclaimer: Please note that your submission will be evaluated against a series of test cases, including some that are kept secret and not provided in the problem statement. It's important to remember that each test case is independent, and your program will be restarted for each one. When reading input, make sure to use the `input()` function. Do not use any string in the `input()` prompt (argument), as this will print the prompt to standard output, which will be interpreted as an incorrect answer. Avoid hardcoding solutions based on the sample cases, as this will not reflect the versatility needed to handle unseen data. Output should be printed directly to standard output, and may be necessary to ensure that your code works efficiently within the given time constraints.

Output

A list of numbers representing the decoded and cleansed message from the scroll.

Sample Input 1	**Sample Output 1**
`1 7 5 9 1 6` `5`	`[1, 7, 9, 1]`

Sample Input 2	**Sample Output 2**
`1 1 1 1 2` `1`	`[2]`

Sample Input 3	**Sample Output 3**
`1 2 3 4 5 6` `10`	`[1, 2, 3, 4, 5, 6]`