Queues with the possibility of penalizing elements

Sometimes, an element in a queue needs to be penalized and moved to the end. In this practice, **modify the given implementation of queues** with linked lists (i.e. "Queue.h" in the Virtual Campus, with also the file "Exceptions.h") to include a **new operation in this ADT**. This operation should take the element in the position given in the parameter (starting counting on zero) and move to the end of the queue, with the following notation:

***void** penalizeElem(**int** position)*

For implementing this operation**, do not create or delete any element or node**, as this could imply performing issues regarding the type used as template. You just need to **change the pointers of the corresponding nodes**.

Implement a file with the main function and whatever other functions you need, to test the new ADT for considering the corresponding inputs and outputs.

Input

The first line will indicate the number of cases. Each case will be defined with a two lines. In each case, the first line is a queue of integers (indicating the number of elements and each element). The second line of each case is the position of the element to remove (starting to count on zero).

Output

The output of each case is printed in one line containing the queue (i.e. each element of the queue followed by a space character)

Example of input

```
3
5 0 1 2 3 4
2
4 7 11 23 47
0
3 5 9 21
2
```

Example of output

```
0 1 3 4 2
11 23 47 7
5 9 21
```