

## 3626 Q

You've got a queue. And you just got to mess with it.

Given a queue of items and a series of queue operations, return the resulting queue.

Queue operations are defined as follows:

*starting* – *position* to *requested* – *position*

meaning one wants the item at the starting position to be moved to the requested position. So if the queue of items were:

*Item1 Item2 Item3 Item4 Item5*

(*Item1* being in position 1, *Item2* in position 2, etc.)

After applying the queue operation: 5 to 2 the resulting queue would be:

*Item1 Item5 Item2 Item3 Item4*

as *Item5* (the item in position 5) was moved to position 2. Multiple queue operations are applied at the same time, however; e.g., given the queue of items:

*Item1 Item2 Item3 Item4 Item5 Item6 Item7 Item8*

If the following queue operations were applied:

2 to 6; 6 to 3; 4 to 5; 5 to 2; 7 to 4; 8 to 1

then the resulting queue would be:

*Item8 Item5 Item6 Item7 Item4 Item2 Item1 Item3*

As you can see, the queue operations are strictly enforced, with other items (not involved in queue operations) maintaining their order and moving to vacant positions in the queue. Note that no two queue operations will have the same *starting-position* or same *requested-position* defined.

### Input

Input to this problem will begin with a line containing a single integer  $x$  indicating the number of datasets. Each data set consists of three components:

1. Start line — A single line, ' $m\ n$ ' ( $1 \leq m, n \leq 20$ ) where  $m$  indicates the number of items in the queue and  $n$  indicates the number of queue operations.
2. Queue items — A line of short (between 1 and 8 characters) alphanumeric names for the items in the queue. Names are unique for a given data set and contain no whitespace.
3. Queue operations —  $n$  lines of queue operations in the format '*starting – position requested – position*'.

### Output

For each dataset, output the queue after the queue operations have been applied. Print the elements of the queue on a single line, starting from the first and ending with the last, with a single space separating each item.

**Sample Input**

```
3
5 1
alpha beta gamma delta epsilon
5 2
8 6
a b c d e f g h
2 6
6 3
4 5
5 2
7 4
8 1
3 2
foo bar baz
3 1
1 3
```

**Sample Output**

```
alpha epsilon beta gamma delta
h e f g d b a c
baz bar foo
```