Problem M. Mistake

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

As an apprentice algorithms enthusiast, it is not a great surprise that Mike struggles to cope with overly complex systems. Unfortunately, this turned out to be a big problem in the company he is currently interning.

Mike's assigned project involves tinkering with the company's Intelligent Cluster for Parallel Computation. This is just a fancy name; in reality, the system is just a simple job scheduler, handling a total of n jobs. Some jobs might depend on successful execution of other jobs before being able to be executed. There are m such dependencies in total.

It is guaranteed that there are no (direct or indirect) circular dependencies between jobs.

When a run is started, the systems intelligently picks an order to execute these jobs so that all the dependencies are met (the order may change between different runs). After picking a valid ordering, it starts executing each of the n jobs in that order. When the system starts executing a job, it prints the id of the job to a log file.

Unfortunately, today was Mike's first day interning at the company and he wasn't very cautious. Consequently, he accidentally ran the system k times in parallel. The system started erratically launching jobs and printing to the log file. Now the log file contains $n \cdot k$ ids of all the jobs that were executed. The job ids from the same run have been printed in the order they were executed, but the outputs from different runs may appear interweaved arbitrarily.

Your task is to figure out which jobs were executed in each of the k runs from the information inside the log file.

Input

The first line of the input will contain three integers n, k, m ($1 \le n, k \le 500\,000$, $0 \le m \le 250\,000$, $n \cdot k \le 500\,000$), the number of jobs in the system, the number of runs Mike had triggered, and the number of dependencies.

The following m lines will contain a pair a_i, b_i ($1 \le a_i, b_i \le n, a_i \ne b_i$, for all $1 \le i \le m$) describing a dependency of kind: "job a_i must be executed before job b_i".

Finally, the last line of the input contains $n \cdot k$ integers c_i ($1 \le c_i \le n$, for all $1 \le i \le n \cdot k$), the job ids that have been printed in the log file, in order.

Output

Output a single line consisting of $n \cdot k$ integers r_i ($1 \le r_i \le k$, for all $1 \le i \le n \cdot k$), the run id corresponding to each of the jobs in the log file. More specifically, r_i should be the run id corresponding to the i-th job, as it appears in the log file.

If multiple solutions are possible, any one is accepted. It is guaranteed that the input data is valid and that a solution always exists.

Example

standard input	standard output

3 3 2
1 2
1 3
1 1 2 3 3 2 1 2 3
	1 2 2 1 2 1 3 3 3