Tasks

After a recent inspiration, Ivancho desided to finaly get to working and do some tasks he has been putting aside for a while. The tasks turned out to be more than expected so he had to make a list and write them down.

While writing the list, Ivancho noticed that there are certain tasks he must do before proceeding to other tasks. He also noticed that some tasks are too complicated even for him so he will have to put them aside for now.

Before he starts working, Ivancho wants to know how many tasks he can do today. Thats why he asks you to help him by writing a program tasks, which by a given list of tasks he has to do and a list of tasks he can't do finds out the number of tasks he can do.

Input: The first row of the input file tasks.in contains two integers: N - the number of tasks on Ivancho's list and M - the number of pairs where one task must precede the other. The next M rows contain two integers each: T1 and T2 - the tasks T2 must be done before the task T1. The indexing starts from 0.

The next row contains one integer K - the number of tasks Ivancho can't do. The next K rows will contain one ingeter - the index of the task he can't do.

Output: The only row in the output file **task.out** should contain one integer - the amount of tasks Ivancho can do.

Limits:
N < 5000
M < 150000
K < 300
The will always exist a sequence in which all tasks can be done.

Time limit: 1 sec.
Memory limit: 256 MB.

Remark: Reading and writing to a file can be done using the appropriate statement. You can use the freopen statement from the standart library <fstream> by adding the flowing two lines at the begining of your main function:

```
freopen ("basek.in", "r", stdin);
freopen ("basek.out", "w" , stdout);
```

Sample test:

tasks.in	tasks.out
9 8 0 3 0 6 1 4 1 6 2 4 3 5 4 7 4 8	4

2 0 8	

Output explanation: Ivancho can't do task 8, but task 8 must be done before task 4, so task 4 can't be done neither. Similarly tasks 1 and 2 can't be done becouse 4 preceedes them. Finally, task 0 can't be done because the test data.