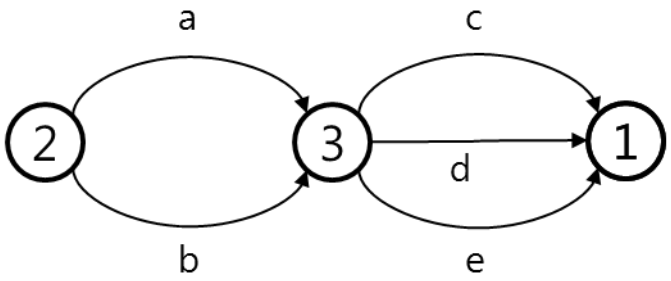
Problem 3

In a search engine, one of the ways of ranking web pages is to calculate how many ways you are able to reach the page. The web consists of N pages and M links that connect two pages. Naturally, a link is directed and there can be multiple links between two pages (even in the same direction). A path is defined to be a way of going from one page to another by using a number of links. Some of the pages are designated as starting points and one page is designated as the end point.

We want to count all possible paths that each uses at most L roads. (A path may use one same link many times and two paths may also share links.)

For example in the figure below (say, 1 is the end point and 2 and 3 are starting points) if L=2, then there are 6 (ac, ad, ae, bc, bd, be) paths that start from 2 and there are 3 paths that start from 3.



Write a program that, given the structure of the web, computes the number of paths that start from each starting point. Because the results can be a very large number, it is required that you print out the remainder when the results are divided by certain integers. Check the input/output conditions below thoroughly.

[Input]

The first line of the input file contains the number T of test cases in the file. In each test case, the first line contains four integers N (the number of pages), M (the number of links), K (the number of starting points), and L (the maximum number of links in a path). ($2 \le N \le 1{,}000$, $1 \le M \le 10{,}000$, $1 \le K \le 10$, $1 \le L \le 10{,}000$) The pages are numbered from 1. Number 1 is the end point and numbers from 2 to K+1 are the starting points. The next M lines each contain a description of a link by two integers s and t, meaning that there is a road from the page numbered s to the page numbered t. The order of the two integers is important and there may exist multiple roads between any two crossings.

There are two kinds of inputs listed as follows.

- Small Set: 2 ≤ N ≤ 20, 1 ≤ M ≤ 500, 1 ≤ K ≤ 10, 1 ≤ L ≤ 5
- Large Set: 2 ≤ N ≤ 1,000, 1 ≤ M ≤ 10,000, 1 ≤ K ≤ 10, 1 ≤ L ≤ 10,000

[Output]

For each test case given, print one line containing K integers. The first one is the remainder when the number of paths from starting point 2 is divided by 2. The second one is the remainder when the number of paths from starting point 3 is divided by 3. The third one is the remainder when the number of paths from starting point 4 is divided by 5. That is, if the crossings 2, 3, 4, 5, 6, …, K+1 are the starting points, you are printing the remainder when the number of paths from each starting point is divided by the first K members in 2, 3, 5, 7, 11, 13, 17, 19, 23, 29. We are using up to 29, because there are only 10 starting points at the most.

[I/O Example]

Input

```
2
3 5 2 2
2 3
2 3
3 1
3 1
3 1
4 4 2 1
2 1
3 2
3 1
4 3
```

Output

```
0 0
1 1
```