Pipes

In the land of the Cahoots, Lomikel is a god of pipes. He governs water pipes, drains, sewers, and maybe even subway tunnels. The Cahoots worship him at numerous sacred springs, which are connected by a huge network of ceremonial pipes. Each pipe directly connects two springs.

On every holiday, the Supreme Plumber (the highest of Lomikel's priests) conducts complicated rituals which involve transport of water through the pipes.

Sometimes, Lomikel's wrath causes a pipe to break, so the Plumber has to use other pipes to make the water flow around the broken pipe. This is not always possible – for some pipes, a different path does not exist. These pipes are called *critical* and the Plumber has to pay special attention to them. You can see critical pipes drawn in bold in the picture below.

Your task is to read a description of the network and find all critical pipes. However, the network is vast and you have only a limited amount of memory. **Memory limit for this task is only 16 MB.**

Input format

The first line of the standard input contains two space-separated integers N and M. Here N is the number of springs ($1 \le N \le 100\,000$) and M is the number of pipes ($1 \le M \le 6\,000\,000$).

Each of the following M lines describes a single pipe. It contains two space-separated integers u and v ($1 \le u, v \le N$) – the springs connected by the given pipe.

Two springs can be connected by multiple pipes, but endpoints of a single pipe are always different springs.

Technical note: It is possible to seek on the standard input (for example to rewind it back to the beginning), but seeking is not necessary to solve the task. Also, reading the input multiple times could be too slow.

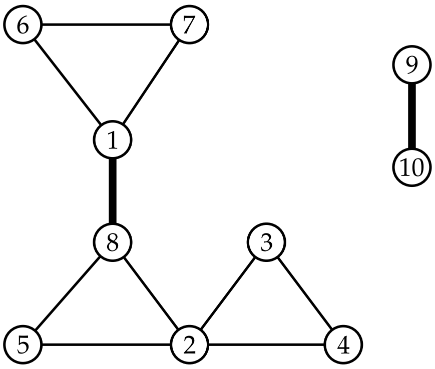
Output format

The standard output consists of a sequence of lines. Each line describes a single critical pipe and it contains two space-separated integers: the endpoints of the pipe.

Critical pipes can be listed in an arbitrary order and so do the endpoints of any single pipe.

Example input

```
10 11
1 7
1 8
1 6
2 8
6 7
5 8
2 5
2 3
2 4
3 4
10 9
```



Example output

```
1 8
9 10
```

Grading

There are 10 groups of tests, each of them worth 10 points. The upper bounds on all parameters in each group of tests are as follows.

Group	1	2	3	4	5	6	7	8	9	10
Limit on N	100	5 000	4 000	10 000	30 000	70 000	80 000	100 000	100 000	100 000
Limit on M	200	15 000	600 000	1 200 000	1 500 000	2 000 000	3 000 000	4 000 000	5 000 000	6 000 000