Suppose IIITG has introduced a new social networking site for its students. There are n students, and hence, n users on the website. Let U be the set of all users. The relationship on the website is bi-directional, i.e., if $u_1 \in U$ is a *friend* of $u_2 \in U$ then, u_2 is a *friend* of u_1. You have created a malware called EndSem. EndSem is a Trojan horse and a virus. You have shared the link of EndSem with all the users of the website. Therefore, any user may download it. The malware has the following properties:

1. Suppose a user $u \in U$ downloads it. Then u becomes the first victim of the malware.

2. If user u is a victim of the malware, then the credentials (the username and the password) of u is compromised.

3. If u is a victim of EndSem, then it (EndSem) attempts to infect exactly one friend of u (chosen randomly).

4. If EndSem attempts to infect a user who is already a victim of EndSem and is not the first victim, it (EndSem) destroys all instances of itself (EndSem).

5. If EndSem reinfects the first victim, then it sends the credentials of all the victims to you and destroys all instances of itself.

You are not interested in figuring out who downloaded the malware. Therefore, when u_1 is the first victim and the malware traverses in the sequences $\langle u_1, u_2, \ldots, u_{i-1}, u_i, u_{i+1}, \ldots, u_{p-1}, u_p, u_1 \rangle$ is the same as the case when u_i is the first victim and the malware traverse in the sequence $\langle u_i, u_{i+1}, \ldots, u_{p-1}, u_p, u_1, u_2, \ldots, u_{i-1}, u_i \rangle$ to you. Note that if u_1 is the first victim, then the malware traverse sequences $\langle u_1, u_2, \ldots, u_{p-1}, u_p, u_1 \rangle$ is same as the sequence $\langle u_1, u_p, u_{p-1}, \ldots, u_2, u_1 \rangle$, i.e., a sequence and it's reverse are the same. Assuming only one user will download the malware, write a program to count the number of distinct cases in which you may get the credentials of the users. You also need to print the distinct cases.

The input to your program is stored in a file. Your program must read the file to process the input. You need to pass the input filename as a command-line argument. Sample input is provided in the file Users.txt. You need to execute your program for that input file and upload the output (in a separate file) while submitting the code.

The input file has the following structure. The first two entries indicate the number of users on the website and the number of pairs of friends, respectively. Then each pair of entries indicate that these two users are friends with each other. We assume that the 0 is the id of the first user, and all the users are assigned a unique number from the set $\{0, 1, 2, \ldots, n-1\}$. To realize the structure consider the following input in an input file:

```
7 5
0 1
1 2
2 3
3 0
1 6
```

It means that there are 7 users and 5 pairs of friends. User 0 is a friend of user 1 and vice versa, user 1 is a friend of user 2 and vice versa, user 2 is a friend of user 3 and vice versa, user 3 is a friend of user 0 and vice versa, and user 1 is a friend of user 6 and vice versa. Therefore, the expected output is 1 as only an equivalent sequence of $\langle 0, 1, 2, 3, 0 \rangle$ is possible. You also need to print a sequence that represents the case $\langle 0, 1, 2, 3, 0 \rangle$, for instance, you may print

```
0 1 2 3
```

to represent the case. Note that, $\langle 1, 3, 2, 0, 1 \rangle$ is not a valid sequence for this case.

Marking scheme:
File read: 15 marks.
Finding one sequence: 20 marks.
Counting all sequences: 20.
Printing all sequences: 20.
Correct answer for Users.txt: 20 marks.
Coding style, naming convention: 5 marks.