

Molecules

A software company is currently testing its new chemistry drawing tool. The tool automatically produces a sample containing some number of artificially constructed molecules. To verify the output of the tool, the company has to count how many different molecule structures are contained in the sample.

Two molecules have the same structure if there is a bijection among their atoms which preserves the bonds. The bijection does not take into account to what element a particular atom belongs, only the molecule structure is relevant.

The structure of each molecule in the sample is limited by the following rules.

- Each atom is bonded to one or three other atoms.
- The only cycles appearing in the molecule structure are triangles (cycles of length three).

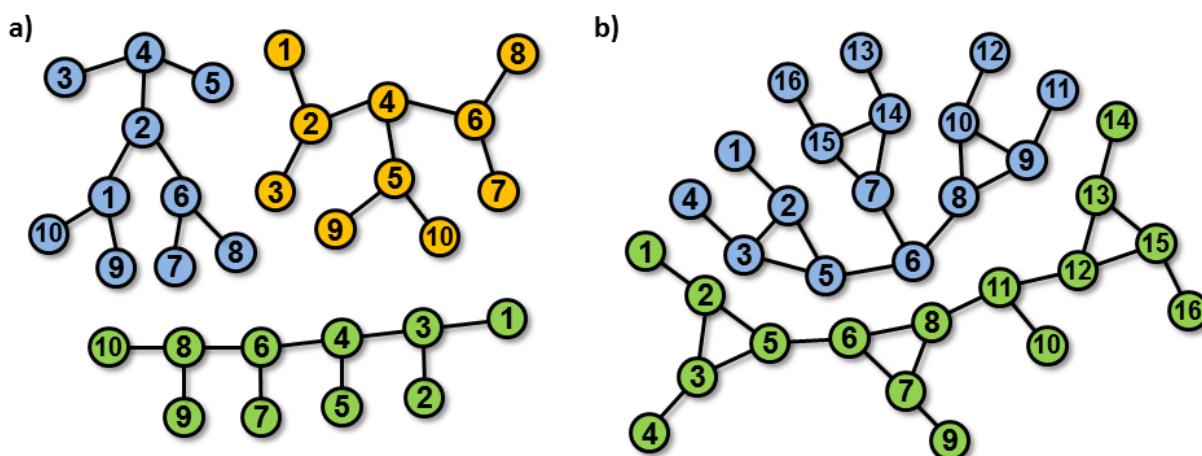


Figure 1. a) A sample containing 3 molecules. Circles represent atoms, edges represent bonds. The blue and orange molecule have the same structure, the green molecule is different. b) A sample containing 2 molecules, both having different structure.

The task

Given a sample of molecules, the task is to classify their structure and print out the counts of molecules for each detected structural class.

Input

The first line of the input is formed of three integers A , B , M separated by space. M is the number of molecules in the sample. Each molecule is composed of A atoms and B bonds between the atoms. Next, there are M lists, each of them giving B bonds for one of the molecules. Every list consists of B lines where each line contains integers A_1 , A_2 separated by space. This pair says that there is a bond between the atoms A_1 and A_2 . The bonds are listed in a random order. The atoms are numbered from 1 to A .

It holds $A \leq 7000$, $B \leq 7000$ and $M \leq 250$.

Output

The output is one line containing N integers C_1, \dots, C_N separated by space. The integers are arranged in the ascending order. N is the total number of structural classes detected among the molecules. Each C_i is the number of molecules with the same structure belonging to the i -th class.

Example 1

Input

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

Output

```
1 2
```

Example 2

Input

```
16 18 2  
1 2  
2 3  
3 4  
2 5  
3 5  
5 6  
6 7  
7 8  
8 6  
7 9  
8 11  
11 10  
11 12  
15 12  
15 16  
15 13  
13 14  
13 12  
6 5  
5 2  
5 3  
2 3  
1 2  
3 4  
16 15  
15 14  
14 13  
7 15  
7 14  
7 6  
8 10  
8 6  
9 8  
9 11  
10 9  
10 12
```

Output

```
1 1
```

Example 3

Input

```
8 8 3  
1 3  
3 2  
3 4  
4 5  
5 7  
7 4  
7 8  
5 6  
2 4  
4 3  
4 1  
3 1  
1 5  
3 7  
7 6  
7 8  
1 7  
7 5  
5 2  
2 7  
2 6  
3 5  
3 8  
4 3
```

Output

```
3
```

The data of Example 1 are depicted in Figure 1 a), the data of Example 2 are depicted in Figure 1 b).

Public data

The public data set is intended for easier debugging and approximate program correctness checking. The public data set is stored also in the upload system and each time a student submits a solution it is run on the public dataset and the program output to stdout and stderr is available to him/her.

[Link to public data set](#)