

Breaking Up is Hard to Do

Purpose: Backtracking

Due: April 19th

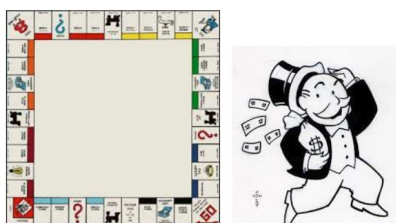


Figure 1: A Monopoly Game Board

The federal trade commission, is tasked with promoting competition by stopping the creation and breaking up monopolies. A monopoly is when a single firm unreasonably restrains competition. When a monopolistic organization is brought before the federal trade commission, the commission attempts to break up the company into smaller companies. These smaller components are designed not to have sufficient market share to dominate a market or even a significant fraction of a market. However if the break up is too severe, the pieces of the original company will not be able to survive. Thus, they have to strike a balance between viability and dominance.

In order to more efficiently operate, the federal trade commission has hired you to create software to help break up companies. The input will be a set of components numbered 1 to n and a target number of sub companies k . Additionally for each component, there will be a list of synergistic components, which should not be combined together to form a new sub company since that sub company will still dominate a market. Your job is to decide how to break up the companies or if no such break up is feasible.

Input

The first line of input will hold 2 integers n and k indicating the number of components and the number of target sub companies. The next n lines hold information about which companies should **not** be combined together. If j appears on the i^{th} line, then i and j should not be combined together. The end of each line will contain a zero.

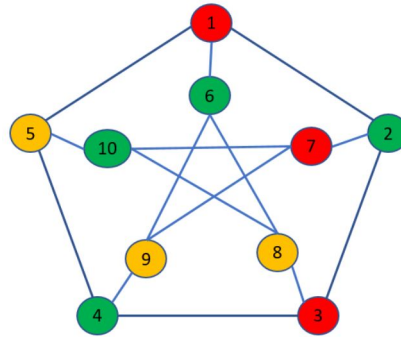


Figure 2: A company with 10 components broken up into 3 sub companies. Edges indicate that those components should not be combined.

Output

The output should be either 1 line with the phrase **no solution** on it, or k lines. Each line should hold the sub companies. Each line will be in alphabetical order and the lines will be ordered in lexicographical order.

Sample Input

```
10 3
5 6 2 0
1 7 3 0
2 8 4 0
5 9 3 0
4 10 1 0
1 9 8 0
2 10 9 0
6 10 3 0
4 6 7 0
5 7 8 0
```

Sample Output

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

How the program will be graded

Memo

| What | pts |
|--|-----|
| Name | 1 |
| Time Analysis $O()$ of every function ¹ (in terms of the number of components) | 6 |
| Space Analysis $O()$ of every function | 6 |
| Test Plan ² with at least 4 original nontrivial tests ³ | 12 |

Source Code Document

| What | pts |
|-----------------------------|-----|
| Name | 1 |
| Description ⁴ | 4 |
| Style | 10 |
| Functionality using the STL | 60 |

¹The main() is a function.

²A test plan is a table with 4 columns and 1 row per test. The columns are named Reason for the test, actual input data, expected output data, and actual output. You do NOT have to have a working program to write a test plan. Each reason should be unique.

³A non trivial test contains only legal data (data that conforms to the input specification) with graphs contain at least 1 vertex .

⁴The description should be written to someone who knows NOTHING about the program. It should discuss what the program does (in your own words). After reading the description the user should be able to create legal input and predict the output.