Problem 2 - $\mathcal{P}$ kingdom (13 points + Bonus 2 points)

Problem Description

$\mathcal{P}$ kingdom, a prosperous country, was established by the wisdom of the ancestors a long time ago.

This country consists of N cities indexed from 1 to N and M directed roads. Each road e_i connects two different cities u_i and v_i, indicating that you can visit city v_i from city u_i.

As a resident in the $\mathcal{P}$ kingdom, you find that there are specific cities where residents may not be able to visit all other cities in the $\mathcal{P}$ kingdom.

In order to meet everyone's daily needs, you have to solve this problem by building new directed roads. Now you can propose a list of roads that need to be built and minimize the number of roads on the list.

Input

The first line contains two integers T, representing the number of test cases, and $flag$ ($flag \in \{0, 1\}$), which will be described in the output section.

Each test case consists of two parts: the first line contains two space-separated integers N, M, representing the number of cities and the number of roads, respectively.

Each line of the following M contains two space-separated integers u_i and v_i, indicating a road connection from u_i to v_i.

- $1 \leq N \leq 1000$
- $0 \leq M \leq \min(1000, N \times (N-1))$
- $1 \leq u_i \leq N$
- $1 \leq v_i \leq N$
- $u_i \neq v_i$

It is guaranteed that the sum of N does not exceed 3000 and the sum of M does not exceed 3000.

Test Group 0 (0 %)

- Sample Input.

Test Group 1 (30 %)

- $flag = 0$
- $u_i < v_i$

Test Group 2 (30 %)

- $flag = 0$

Test Group 3 (20 %)

- $u_i < v_i$

Test Group 4 (20 %)

- No additional constraints.

Bonus Group (2 points)

- $N, M \leq 100000$
- The sum of N does not exceed 100000 and the sum of M does not exceed 100000.

Output

For each testcase, print an integer on the first line representing the number of roads K that need to be built.

If the $flag$ mentioned in the input section is equal to 1, please furthermore print the following K lines: each line contains s_i and t_i, representing building a road from s_i to t_i in the list. If there are multiple ways to match this condition, you can print any of them.

Sample Input 1

```
1 0
4 2
1 2
3 4
```

Sample Output 1

```
2
```

Sample Input 2

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

Sample Output 2

```
3
2 3
4 5
5 1
0
```