

Problem Jul 30: Friendship Day

Time limit: 4 seconds

Recently at the greeting card company, some teams have been reorganized. Because your boss is one of those weirdos who think get-to-know-you games are a good idea, they decided to have you play one of these games for team building in light of the *International Friendship Day*.

Your boss also likes to delegate their half-baked ideas, and this time they chose you to work it out.

In the game, each team member is only allowed to move on a straight line and people have to talk when they bump into each other. By the nature of the game, everyone should be able to bump into everyone they don't know. Also, because your boss does not want you to have *too much* fun, people that do already know each other should not have the possibility to talk.

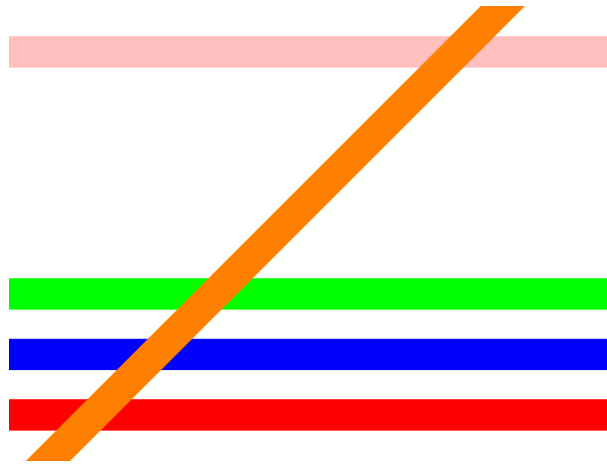


Figure Jul 30.1: Illustration of Sample Output 2.

The problem is only that you now have to find a line for each team member such that your boss's requirements are met.

Input

The input consists of:

- One line with two integers n and m ($1 \leq n \leq 3 \cdot 10^5$, $0 \leq m \leq 3 \cdot 10^5$), the number of team members and the number of pairs of team members that do not know each other.
- m lines, each containing two integers a and b ($1 \leq a, b \leq n$, $a \neq b$) indicating that team members a and b do not know each other.

No pair of team members is given more than once.

Output

If it is not possible to find lines satisfying the given constraints, output "impossible". Otherwise, output "possible", followed by n descriptions of lines. For each line, output the coordinates x_1 , y_1 , x_2 and y_2 of two distinct points (x_1, y_1) and (x_2, y_2) on the line. No coordinate may have an absolute value greater than 10^9 and no two lines may be identical. If there are multiple valid solutions, you may output any one of them.

Sample Input 1

```
3 3
1 2
2 3
3 1
```

Sample Output 1

```
possible
0 0 999 1000
42 43 15 17
3 4 5 6
```

Sample Input 2

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

Sample Output 2

```
possible
10 10 11 11
10 10 20 10
11 11 31 11
12 12 41 12
42 16 1024 16
```

Sample Input 3

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

Sample Output 3

```
impossible
```