Keep Calm and Maintain Social Distancing

Input file:	`standard input`
Output file:	`standard output`
Time limit:	C++/Java/Python: 2/4/20 seconds
Memory limit:	1024 megabytes
Points available:	100

A prestigious tech conference is being held next week, and you've just received your invitation to attend and network with some of the biggest names in software development! It also just so happens that you've recently developed a new software app which you'd like to personally advertise to as many people as possible at the conference, although you quickly realise that you'll only have time to meet with K other people.

Unfortunately, the year is also 2020, and everyone is required to maintain social distancing due to this nasty virus going around. To most effectively market your app, you decide to strategically spend your time meeting with K participants who are the most isolated from each other. You therefore decide to write a program to help you accomplish this task.

There are N participants in total (labelled from 1 to N), and it's well-known that each participant has **no more than two** friends. It's also well-known that friendship is mutual (so if A is friends with B, then B is friends with A). Given a list of all friendships between the participants, we define the *social distance*, denoted "$\text{dist}(A, B)$", between two participants A and B as the minimum number of social connections joining A to B. For example, if A and B are friends, then $\text{dist}(A, B) = 1$; if they are not friends but have a friend in common then $\text{dist}(A, B) = 2$, and so on. If no possible connection exists between A and B, we define $\text{dist}(A, B)$ to be infinite.

Furthermore, we can extend the concept of social distance to larger groups. We define the social distance of a set S of people as the smallest $\text{dist}(A, B)$ for any distinct A and B in the set S. For example, if A, B, C are three distinct participants, then

$$\text{dist}(\{A, B, C\}) = \min\big(\text{dist}(A, B),\ \text{dist}(A, C),\ \text{dist}(B, C)\big).$$

Given some positive integer $K \le N$, your task is to find a subset S of K participants, such that $\text{dist}(S)$ is maximal (i.e. the minimal social distance between any pair of the K participants is maximised).

There may certainly be multiple possible solutions. You may output any valid choice of K participants which yields an optimal solution.

Input

The first line consists of three space-separated integers N ($2 \le N \le 1\,000\,000$), the number of participants, M ($0 \le M \le N$), the number of friendships among the participants, and K ($2 \le K \le N$), the number of participants you wish to meet.

The next M lines consists of two space-separated integers a_i, b_i on each line denoting a friendship between a_i and b_i. It is guaranteed that $a_i \ne b_i$ and that no friendship will be given more than once.

Output

You must output two lines.

The first line should contain a single integer, the minimum distance between any two of the K chosen participants. If it's possible to choose K participants each completely isolated from each other, you must instead output the single word "`infinity`".

The second line should contain the labels of the K chosen participants in any order.

Scoring

For each subtask, a correct solution will score 100% of the points available, otherwise if only the first line of output (the optimal minimum distance) is correct, you will score 40% of the points available. A solution with the incorrect minimum distance will score 0%.

Subtask 1: (0 points) Examples.

Subtask 2: (8 points) $N \leq 15$

Subtask 3: (5 points) $K = 2$

Subtask 4: (6 points) Each participant has at most one friend.

Subtask 5: (6 points) Each participant has exactly two friends, and every pair of participants are socially connected in some way.

Subtask 6: (10 points) Each participant has exactly two friends.

Subtask 7: (15 points) $N \leq 100$

Subtask 8: (20 points) $N \leq 5000$

Subtask 9: (30 points) $N \leq 1\,000\,000$

Examples

standard input	standard output
6 5 3 1 5 2 3 3 6 5 4 2 6	2 1 4 2
6 6 3 1 5 2 3 3 6 5 4 2 6 1 4	1 1 2 3
3 1 2 2 3	infinity 1 2

Note

In the first example, we have that 1 is friends with 5 and 5 is friends with 4. Since 1 is not friends with 4, this means that $\text{dist}(1, 4) = 2$. Furthermore, we have that 2 has no social connection to 1 or 4. Therefore the set of participants $\{1, 2, 4\}$ has a social distance of two, and is optimal because there does not exist a set of three participants having social distance greater than two (note that $\{1, 3, 4\}$ and $\{1, 4, 6\}$ are also valid choices of participants).

The second example is similar to the first, except now 1 is also friends with 4. This forces any set of three participants to have social distance one, so outputting any set of three distinct participants would be a valid solution.

In the third example, only 2 and 3 are friends with each other. Hence the participants $\{1, 2\}$ have no social connections, and so this set has infinite social distance.