

Problem A. Cóp và Khâu

Time Limit 1000 ms

Mem Limit 524288 kB

Xét đơn đồ thị vô hướng $G = (V, E)$ có N ($1 \leq N \leq 10000$) đỉnh và M ($1 \leq M \leq 50000$) cạnh. Người ta định nghĩa một đỉnh gọi là khớp nếu như xoá đỉnh đó sẽ làm tăng số thành phần liên thông của đồ thị. Tương tự như vậy, một cạnh được gọi là cầu nếu xoá cạnh đó sẽ làm tăng số thành phần liên thông của đồ thị.

Vấn đề đặt ra là cần phải đếm tất cả các khớp và cầu của đồ thị G .

Input

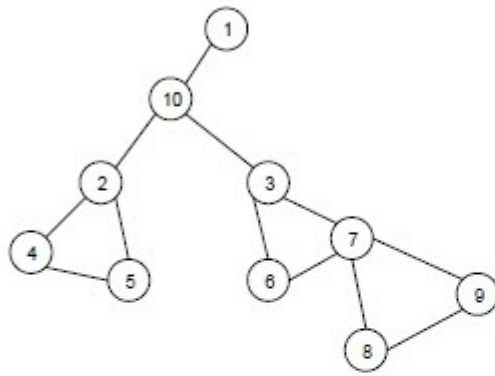
- Dòng đầu: chứa hai số tự nhiên N, M .
- M dòng sau mỗi dòng chứa một cặp số (u, v) (u khác v , $1 \leq u \leq N$, $1 \leq v < N$) mô tả một cạnh của G .

Output

- Gồm một dòng duy nhất ghi hai số, số thứ nhất là số khớp, số thứ hai là số cầu của G

Input	Output
10 12 1 10 10 2 10 3 2 4 4 5 5 2 3 6 6 7 7 3 7 8 8 9 9 7	4 3

Note



Problem B. How to Tourist

Time Limit 3000 ms

OS Linux

Statement [PDF](#)

Given an undirected connected map of N locations (places) labeled by unique names, you must find all *camera locations*.

A camera is placed at a location whose removal (and all incident routes) increases the number of connected components in the remaining map—that is, a location through which any path between at least one pair of other locations must pass.

Input

The input contains multiple test cases. Each test case begins with an integer N ($2 < N \leq 100$), the number of locations.

Then follow N distinct place names (one per line), each a lowercase string of length 1–30.

Next comes a non-negative integer R , the number of bidirectional routes.

Then follow R lines, each with two place names separated by a space, indicating a direct route between them.

There are no self-loops or duplicate routes, and all names are valid.

The input terminates with a line containing $N = 0$, which should not be processed.

Output

For each test case, output:

City map # d : c camera(s) found

where d is the test case number (starting from 1) and c is the number of camera locations. Then print the c place names (one per line) in ascending alphabetical order. Print a blank line between consecutive test case outputs.

Constraints

- $2 < N \leq 100$
- $0 \leq R \leq \frac{N(N-1)}{2}$
- Place names are lowercase latin strings, length 1–30.
- The map is initially connected.

Example

Input	Output
6 sugarloaf maracana copacabana ipanema corcovado lapa 7 ipanema copacabana copacabana sugarloaf ipanema sugarloaf maracana lapa sugarloaf maracana corcovado sugarloaf lapa corcovado 5 guanabarabay downtown botanicgarden colombo sambodromo 4 guanabarabay sambodromo downtown sambodromo sambodromo botanicgarden colombo sambodromo 0	City map #1: 1 camera(s) found sugarloaf City map #2: 1 camera(s) found sambodromo

Problem C. Yap cây là yâp cáp

Time Limit 3000 ms

OS Linux

Statement [PDF](#)

You are given an undirected connected network of N places numbered $1, 2, \dots, N$. Each place has a telephone exchange, and lines connect pairs of exchanges bidirectionally.

A place is called *critical* if, when its exchange fails (is removed), the network becomes disconnected (i.e., there exists at least one pair of remaining places that can no longer reach each other).

Input

The input consists of multiple test cases. Each test case begins with an integer N ($1 \leq N < 100$), the number of places. This is followed by up to N lines describing the network:

- Each line starts with an integer u (the place number), followed by zero or more integers $v_1 v_2 \dots v_k$, indicating that there is a direct line between place u and each v_i .
- All lines of this block collectively describe every connection at least once; connections are bidirectional.
- A line containing a single **0** ends the current test case.

The input terminates with a test case where $N = 0$, which should not be processed.

Output

For each test case (except the terminating one), print a single integer: the number of critical places in the network.

Constraints

- $1 \leq N < 100$
- The network is initially connected.
- Place numbers and adjacency lists contain integers between 1 and N .

Example

Input	Output
5 5 1 2 3 4 0 6 2 1 3 5 4 6 2 0 0	1 2

Problem D. GTA Vice city

Time Limit	1000 ms
Mem Limit	1572864 kB
Code Length Limit	50000 B
OS	Linux

Vice City được xây dựng trên một nhóm đảo, với các cây cầu nối giữa chúng. Như bất kỳ ai ở Vice City đều biết, nỗi sợ lớn nhất của người dân nơi đây là một ngày nào đó các đảo sẽ bị ngập chìm. Vấn đề lớn là khi các đảo bị ngập, một số đảo khác có thể bị mất kết nối. Bạn đã được thị trưởng Vice City thuê để cho ông ấy biết có bao nhiêu đảo, khi bị ngập, sẽ làm mất kết nối các phần của Vice City. Bạn cần biết rằng ban đầu tất cả các đảo trong thành phố đều được kết nối với nhau.

Đầu vào

Đầu vào sẽ bao gồm một loạt các trường hợp kiểm tra. Mỗi trường hợp bắt đầu với số N ($1 \leq N \leq 10^4$) là số đảo, và số M là số cầu ($1 \leq M \leq 10^5$). Tiếp theo sẽ có M dòng, mỗi dòng mô tả một cây cầu. Mỗi dòng trong số M dòng này sẽ chứa hai số nguyên U_i, V_i ($1 \leq U_i, V_i \leq N$), chỉ ra rằng có một cây cầu nối đảo U_i và V_i . Đầu vào kết thúc khi $N = M = 0$.

Đầu ra

Với mỗi trường hợp trong đầu vào, bạn phải in ra một dòng chỉ ra số đảo mà khi bị ngập sẽ làm mất kết nối các phần của thành phố.

Ví dụ

Input	Output
3 3 1 2 2 3 1 3 6 8 1 3 6 1 6 3 4 1 6 4 5 2 3 2 3 5 0 0	0 1

Problem E. Họa sĩ người Áo

Time Limit 2000 ms

Mem Limit 262144 kB

Input File stdin

Output File stdout

Bạn có một $n \times m$ tờ giấy kẻ ô vuông. Một số ô vuông trên đó đã được tô màu. Hãy ký hiệu tập hợp tất cả các ô được tô màu là A . Tập hợp A này đang liên thông. Nhiệm vụ của bạn là tìm số lượng ô vuông tối thiểu cần xóa khỏi tập hợp A để làm cho nó không còn liên thông nữa.

Một tập hợp các ô được tô màu được gọi là *liên thông* nếu với mọi cặp ô a và b trong tập hợp này, tồn tại một chuỗi các ô bắt đầu từ a và kết thúc ở b sao cho trong chuỗi này, mọi ô (trừ ô cuối cùng) đều có chung một cạnh với ô tiếp theo trong chuỗi. Theo định nghĩa, tập hợp rỗng và tập hợp chỉ chứa đúng một ô vuông luôn được coi là liên thông.

Đầu vào

Dòng đầu tiên chứa hai số nguyên cách nhau bởi dấu cách n và m ($1 \leq n, m \leq 50$) — kích thước của tờ giấy.

Mỗi dòng trong số n dòng tiếp theo chứa m ký tự — mô tả tờ giấy: ký tự thứ j của dòng thứ i bằng "#" nếu ô tương ứng được tô màu (thuộc tập hợp A), hoặc bằng "." nếu ô tương ứng không được tô màu (không thuộc tập hợp A). Đảm bảo rằng tập hợp tất cả các ô được tô màu A là liên thông và không rỗng.

Đầu ra

Ở dòng đầu tiên, in ra số lượng ô vuông tối thiểu cần xóa để làm cho tập hợp A không liên thông. Nếu không thể thực hiện, hãy in -1.

Ví dụ

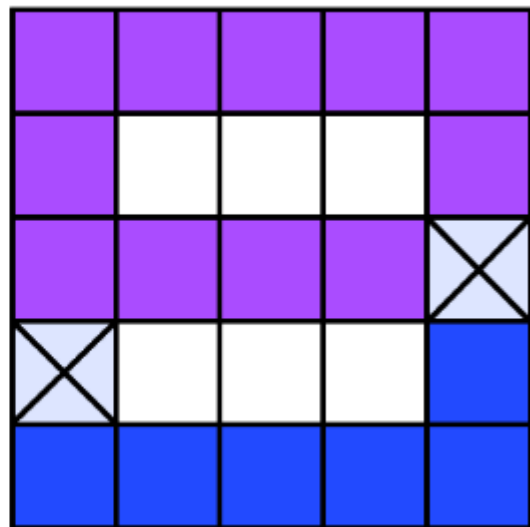
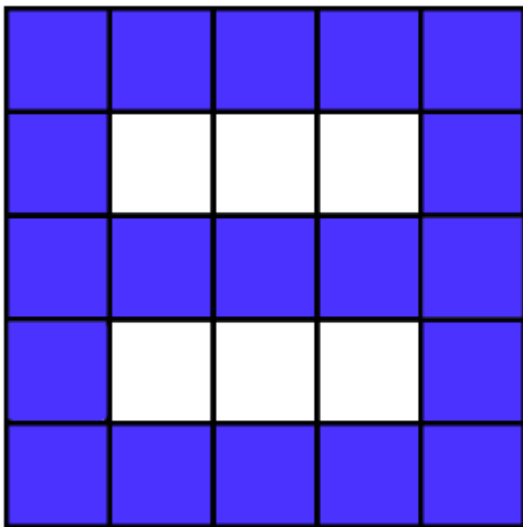
Input	Output
5 4 #### #..# #..# #..# ####	2

Input	Output
5 5 ##### #...# ##### #...# #####	2

Ghi chú

Trong ví dụ đầu tiên, bạn có thể xóa bất kỳ hai ô nào không chung cạnh. Sau đó, tập hợp các ô được tô màu sẽ không còn liên thông nữa.

Ghi chú cho ví dụ thứ hai được minh họa trong hình dưới đây. Bên trái là hình ảnh tập hợp ô ban đầu. Bên phải là tập hợp sau khi xóa các ô. Các ô bị xóa được đánh dấu bằng dấu chéo.



In a computer network a link L, which interconnects two servers, is considered critical if there are at least two servers A and B such that all network interconnection paths between A and B pass through L. Removing a critical link generates two disjoint sub-networks such that any two servers of a sub-network are interconnected. For example, the network shown in figure 1 has three critical links that are marked bold: 0 - 1, 3 - 4 and 6 - 7.

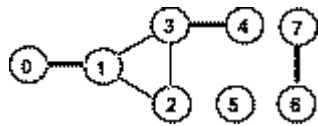


Figure 1: Critical links

- It is known that:
1. the connection links are bi-directional;
 2. a server is not directly connected to itself;
 3. two servers are interconnected if they are directly connected or if they are interconnected with the same server;
 4. the network can have stand-alone sub-networks.

Write a program that finds all critical links of a given computer network.

Input

The program reads sets of data from a text file. Each data set specifies the structure of a network and has the format:

```
no_of_servers
server0 (no_of_direct_connections) connected_server ... connected_server
...
serverno_of_servers (no_of_direct_connections) connected_server ... connected_server
```

The first line contains a positive integer *no_of_servers*(possibly 0) which is the number of network servers. The next *no_of_servers* lines, one for each server in the network, are randomly ordered and show the way servers are connected. The line corresponding to *server_k*, $0 \leq k \leq no_of_servers - 1$, specifies the number of direct connections of *server_k* and the servers which are directly connected to *server_k*. Servers are represented by integers from 0 to *no_of_servers* - 1. Input data are correct. The first data set from sample input below corresponds to the network in figure 1, while the second data set specifies an empty network.

Output

The result of the program is on standard output. For each data set the program prints the number of critical links and the critical links, one link per line, starting from the beginning of the line, as shown in the sample output below. The links are listed in ascending order according to their first element. The output for the data set is followed by an empty line.

Sample Input

```
8
0 (1) 1
1 (3) 2 0 3
2 (2) 1 3
3 (3) 1 2 4
4 (1) 3
7 (1) 6
6 (1) 7
5 (0)
```

```
0
```

Sample Output

```
3 critical links
0 - 1
3 - 4
6 - 7
```

```
0 critical links
```

According to the Automobile Collision Monitor (ACM), most fatal traffic accidents occur on two-way streets. In order to reduce the number of fatalities caused by traffic accidents, the mayor wants to convert as many streets as possible into one-way streets. You have been hired to perform this conversion, so that from each intersection, it is possible for a motorist to drive to all the other intersections following some route.

You will be given a list of streets (all two-way) of the city. Each street connects two intersections, and does not go through an intersection. At most four streets meet at each intersection, and there is at most one street connecting any pair of intersections. It is possible for an intersection to be the end point of only one street. You may assume that it is possible for a motorist to drive from each destination to any other destination when every street is a two-way street.

Input

The input consists of a number of cases. The first line of each case contains two integers n and m . The number of intersections is n ($2 \leq n \leq 1000$), and the number of streets is m . The next m lines contain the intersections incident to each of the m streets. The intersections are numbered from 1 to n , and each street is listed once. If the pair $i\ j$ is present, $j\ i$ will not be present. End of input is indicated by $n = m = 0$.

Output

For each case, print the case number (starting from 1) followed by a blank line. Next, print on separate lines each street as the pair $i\ j$ to indicate that the street has been assigned the direction going from intersection i to intersection j . For a street that cannot be converted into a one-way street, print both $i\ j$ and $j\ i$ on two different lines. The list of streets can be printed in any order. Terminate each case with a line containing a single ‘#’ character.

Note: There may be many possible direction assignments satisfying the requirements. Any such assignment is acceptable.

Sample Input

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

Sample Output

```
1

1 2
2 4
3 1
3 6
4 3
5 2
5 4
6 4
6 7
7 5
#
2

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

The Queen of Nlogonia is a fan of mazes, and therefore the queendom’s architects built several mazes around the Queen’s palace. Every maze built for the Queen is made of rooms connected by corridors. Each corridor connects a different pair of distinct rooms and can be transversed in both directions.

The Queen loves to stroll through a maze’s rooms and corridors in the late afternoon. Her servants choose a different challenge for every day, that consists of finding a simple path from a start room to an end room in a maze. A simple path is a sequence of *distinct* rooms such that each pair of consecutive rooms in the sequence is connected by a corridor. In this case the first room of the sequence must be the start room, and the last room of the sequence must be the end room. The Queen thinks that a challenge is good when, among the routes from the start room to the end room, *exactly* one of them is a simple path. Can you help the Queen’s servants to choose a challenge that pleases the Queen?

For doing so, write a program that given the description of a maze and a list of queries defining the start and end rooms, determines for each query whether that choice of rooms is a good challenge or not.

Input

Each test case is described using several lines. The first line contains three integers R , C and Q representing respectively the number of rooms in a maze ($2 \leq R \leq 10^4$), the number of corridors ($1 \leq C \leq 10^5$), and the number of queries ($1 \leq Q \leq 1000$). Rooms are identified by different integers from 1 to R . Each of the next C lines describes a corridor using two distinct integers A and B , indicating that there is a corridor connecting rooms A and B ($1 \leq A < B \leq R$). After that, each of the next Q lines describes a query using two distinct integers S and T indicating respectively the start and end rooms of a challenge ($1 \leq S < T \leq R$). You may assume that within each test case there is at most one corridor connecting each pair of rooms, and no two queries are the same.

The last test case is followed by a line containing three zeros.

Output

For each test case output $Q + 1$ lines. In the i -th line write the answer to the i -th query. If the rooms make a good challenge, then write the character ‘Y’ (uppercase). Otherwise write the character ‘N’ (uppercase). Print a line containing a single character ‘-’ (hyphen) after each test case.

Sample Input

6 5 3
1 2
2 3
2 4
2 5
4 5
1 3
1 5
2 6
4 2 3
1 2
2 3
1 4
1 3
1 2
0 0 0

Sample Output

Y
N
N
-
N
Y
Y
-

Problem I. Case of Computer Network

Time Limit 3000 ms

Mem Limit 262144 kB

Andrewid Android là một thám tử nổi tiếng trong thiên hà. Bây giờ anh ấy đang chuẩn bị phòng thủ chống lại một cuộc tấn công có thể từ hacker vào một mạng máy tính lớn.

Trong mạng này có n đỉnh, một số cặp đỉnh được kết nối bởi m kênh không hướng. Dự định là chuyển q tin nhắn quan trọng qua mạng này, tin nhắn thứ i phải được gửi từ đỉnh s_i đến đỉnh d_i thông qua một hoặc nhiều kênh, có thể thông qua một số đỉnh trung gian.

Để bảo vệ chống lại các cuộc tấn công, một thuật toán đặc biệt đã được phát triển. Thật không may là nó chỉ có thể áp dụng cho mạng chứa các kênh hướng. Do đó, vì không thể tạo ra các kênh mới, đã quyết định cho mỗi kênh không hướng hiện có để cho phép chúng truyền dữ liệu chỉ trong một trong hai hướng.

Nhiệm vụ của bạn là xác định xem có thể chọn hướng cho mỗi kênh sao cho mỗi trong số q tin nhắn có thể được truyền thành công hay không.

Nhập

Dòng đầu tiên chứa ba số nguyên n , m và q ($1 \leq n, m, q \leq 2 \cdot 10^5$) — số lượng đỉnh, kênh và tin nhắn quan trọng.

Tiếp theo m dòng chứa hai số nguyên mỗi dòng, v_i và u_i ($1 \leq v_i, u_i \leq n, v_i \neq u_i$), có nghĩa là giữa các đỉnh v_i và u_i có một kênh. Giữa một cặp đỉnh có thể tồn tại nhiều hơn một kênh.

Tiếp theo q dòng chứa hai số nguyên s_i và d_i ($1 \leq s_i, d_i \leq n, s_i \neq d_i$) — số của các đỉnh nguồn và đích của tin nhắn tương ứng.

Không đảm bảo rằng ban đầu có thể truyền tất cả các tin nhắn.

Đầu ra

Nếu có một giải pháp tồn tại, in trên một dòng duy nhất "Có" (không có dấu ngoặc kép). Nếu không, in "Không" (không có dấu ngoặc kép).

Ví dụ 1

Input	Output
4 4 2 1 2 1 3 2 3 3 4 1 3 4 2	Yes

Ví dụ 2

Input	Output
3 2 2 1 2 3 2 1 3 2 1	No

Ví dụ 3

Input	Output
3 3 2 1 2 1 2 3 2 1 3 2 1	Yes

Ghi chú

Trong ví dụ kiểm tra đầu tiên, bạn có thể gán hướng, ví dụ, như sau: $1 \rightarrow 2$, $1 \rightarrow 3$, $3 \rightarrow 2$, $4 \rightarrow 3$. Sau đó, đường dẫn cho tin nhắn đầu tiên sẽ là $1 \rightarrow 3$, và cho tin nhắn thứ hai sẽ là $4 \rightarrow 3 \rightarrow 2$.

Trong ví dụ kiểm tra thứ ba, bạn có thể gán hướng, ví dụ, như sau: $1 \rightarrow 2$, $2 \rightarrow 1$, $2 \rightarrow 3$. Sau đó, đường dẫn cho tin nhắn đầu tiên sẽ là $1 \rightarrow 2 \rightarrow 3$, và cho tin nhắn thứ hai sẽ là $2 \rightarrow 1$.

Problem J. Ngày đẹp trời để nói chia tay

Time Limit 3000 ms

Mem Limit 262144 kB

Lại một lần nữa, Berland đang trải qua những thời khắc khó khăn! Nhiều thị trấn có những căng thẳng đến mức có thể xảy ra nội chiến.

Có n thị trấn ở Reberland, một số cặp thị trấn được kết nối bằng các con đường hai chiều. Không đảm bảo rằng có thể đi từ thị trấn này đến thị trấn khác bằng các con đường này.

Thị trấn s và t tuyên bố chấm dứt mọi quan hệ và dự định loại bỏ khả năng di chuyển giữa chúng qua các con đường. Hiện tại có thể cần đóng một vài con đường sao cho việc di chuyển từ s đến t bằng đường bộ trở nên không thể. Mỗi thị trấn đồng ý chi trả chi phí để đóng không quá một con đường, do đó tổng số con đường bị đóng sẽ **không quá hai**.

Hãy giúp họ tìm tập hợp không quá hai con đường sao cho sau khi đóng các con đường này sẽ không còn đường đi giữa s và t . Với mỗi con đường đã ước tính chi phí đóng cửa. Trong tất cả các tập hợp, hãy tìm tập có tổng chi phí đóng cửa thấp nhất.

Đầu vào

Dòng đầu tiên của đầu vào chứa hai số nguyên n và m ($2 \leq n \leq 1000$, $0 \leq m \leq 30\,000$) — số lượng thị trấn ở Berland và số lượng con đường.

Dòng thứ hai chứa hai số nguyên s và t ($1 \leq s, t \leq n$, $s \neq t$) — chỉ số các thị trấn muốn chấm dứt quan hệ.

Sau đó là m dòng, mỗi dòng chứa ba số nguyên x_i, y_i và w_i ($1 \leq x_i, y_i \leq n$, $1 \leq w_i \leq 10^9$) — chỉ số các thị trấn được kết nối bởi con đường thứ i , và ngân sách để đóng con đường đó.

Tất cả các con đường đều hai chiều. Có thể có nhiều hơn một con đường kết nối cùng một cặp thị trấn. Các con đường nối một thị trấn với chính nó cũng được phép.

Đầu ra

Dòng đầu tiên in ra ngân sách tối thiểu cần thiết để chấm dứt quan hệ giữa s và t , nếu được phép đóng không quá hai con đường.

Dòng thứ hai in ra giá trị c ($0 \leq c \leq 2$) — số lượng con đường cần đóng trong giải pháp tìm được.

Dòng thứ ba in ra theo bất kỳ thứ tự nào c số nguyên khác nhau từ 1 đến m — chỉ số các con đường bị đóng. Xem các con đường được đánh số từ 1 đến m theo thứ tự xuất hiện trong đầu vào.

Nếu không thể làm cho các thị trấn s và t bị ngắt kết nối bằng cách đóng không quá 2 con đường, đầu ra chỉ chứa một dòng duy nhất với -1 .

Nếu có nhiều đáp án, bạn có thể in ra bất kỳ đáp án nào trong số đó.

Ví dụ

Input	Output
6 7 1 6 2 1 6 2 3 5 3 4 9 4 6 4 4 6 5 4 5 1 3 1 3	8 2 2 7

Input	Output
6 7 1 6 2 3 1 1 2 2 1 3 3 4 5 4 3 6 5 4 6 6 1 5 7	9 2 4 5

Input	Output
5 4 1 5 2 1 3 3 2 1 3 4 4 4 5 2	1 1 2

Input	Output
2 3 1 2 1 2 734458840 1 2 817380027 1 2 304764803	-1

Problem K. Tourist Reform

Time Limit 4000 ms

Mem Limit 262144 kB

Berland là một đất nước du lịch! Ít nhất là nó có thể trở thành như vậy – chính phủ của Berland tin tưởng vào điều này.

Có n thành phố ở Berland, một số cặp thành phố được kết nối bởi đường cao tốc hai chiều. Mỗi con đường kết nối hai thành phố khác nhau. Ở Berland không có con đường nào kết nối cùng một cặp thành phố. Có thể đi từ bất kỳ thành phố nào đến bất kỳ thành phố nào khác bằng cách sử dụng các con đường hai chiều đã cho.

Theo cải cách, mỗi con đường sẽ trở thành một chiều. Nó sẽ được hướng tới một trong hai hướng.

Để tối đa hóa sự hấp dẫn du lịch của Berland, sau cải cách, cho mỗi thành phố i giá trị r_i sẽ được tính toán. Nó sẽ bằng số lượng thành phố x mà có một đường hướng từ thành phố i đến thành phố x . Nói cách khác, r_i sẽ bằng số lượng thành phố mà có thể đến từ thành phố i bằng đường cao tốc.

Chính phủ chắc chắn rằng sự chú ý của du khách sẽ tập trung vào giá trị tối thiểu của r_i .

Giúp chính phủ Berland thực hiện cải cách để tối đa hóa giá trị tối thiểu của r_i .

Nhập

Dòng đầu tiên chứa hai số nguyên n, m ($2 \leq n \leq 400\,000$, $1 \leq m \leq 400\,000$) – số lượng thành phố và số lượng con đường.

Các dòng tiếp theo mô tả các con đường ở Berland: con đường thứ j chứa hai số nguyên u_j và v_j ($1 \leq u_j, v_j \leq n$, $u_j \neq v_j$), trong đó u_j và v_j là số của các thành phố được kết nối bởi con đường thứ j .

Các thành phố được đánh số từ 1 đến n . Đảm bảo rằng có thể đi từ bất kỳ thành phố nào đến bất kỳ thành phố nào khác bằng cách đi theo các con đường hai chiều. Ở Berland không có con đường nào kết nối cùng một cặp thành phố.

Đầu ra

Trong dòng đầu tiên in một số nguyên duy nhất - giá trị tối đa có thể $\min_{1 \leq i \leq n} \{r_i\}$ sau khi hướng các con đường.

Các dòng tiếp theo phải chứa mô tả của các con đường sau khi hướng: con đường thứ j phải chứa hai số nguyên u_j, v_j , điều đó có nghĩa là con đường thứ j sẽ được hướng từ thành phố u_j đến thành phố v_j . In các con đường theo thứ tự giống như chúng được cung cấp trong dữ liệu nhập.

Ví dụ

Input	Output
7 9 4 3 2 6 7 1 4 1 7 3 3 5 7 4 6 5 2 5	4 4 3 6 2 7 1 1 4 3 7 5 3 7 4 5 6 2 5

Problem L. Non-academic Problem

Time Limit 2000 ms

Mem Limit 262144 kB

Bạn được cho một đồ thị vô hướng liên thông, các đỉnh được đánh số bằng số nguyên từ 1 đến n . Nhiệm vụ của bạn là tối thiểu hóa số cặp đỉnh $1 \leq u < v \leq n$ mà giữa chúng tồn tại một đường đi trong đồ thị này. Để đạt được điều này, bạn có thể loại bỏ chính xác một cạnh từ đồ thị.

Tìm số cặp đỉnh nhỏ nhất!

Nhập

Mỗi bài kiểm tra bao gồm một số lượng tập dữ liệu đầu vào. Dòng đầu tiên chứa một số nguyên duy nhất t ($1 \leq t \leq 10^4$) — số lượng tập dữ liệu đầu vào. Tiếp theo là mô tả của chúng.

Dòng đầu tiên của mỗi tập dữ liệu đầu vào chứa hai số nguyên n và m ($2 \leq n \leq 10^5$, $n - 1 \leq m \leq \min(10^5, \frac{n \cdot (n-1)}{2})$) — số lượng đỉnh trong đồ thị và số lượng cạnh.

Mỗi trong m dòng tiếp theo chứa hai số nguyên u và v ($1 \leq u, v \leq n$, $u \neq v$), cho biết rằng có một cạnh vô hướng trong đồ thị giữa các đỉnh u và v .

Đảm bảo rằng đồ thị cho trước là liên thông và không có nhiều cạnh.

Đảm bảo rằng tổng của n và tổng của m trên tất cả các tập dữ liệu đầu vào không vượt quá $2 \cdot 10^5$.

Đầu ra

Đối với mỗi tập dữ liệu đầu vào, đầu ra số cặp đỉnh nhỏ nhất có thể đến được, nếu chính xác một cạnh có thể được loại bỏ.

Ví dụ

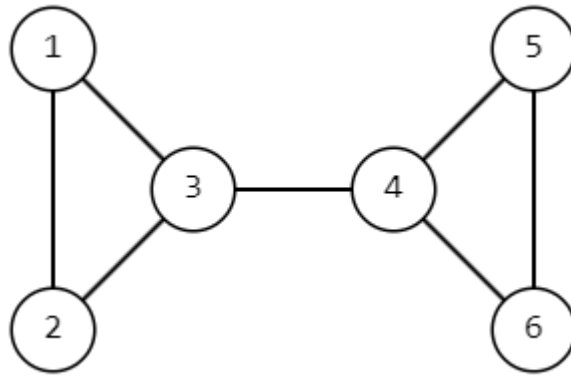
Input	Output
6	0
2 1	3
1 2	4
3 3	6
1 2	6
2 3	21
1 3	
5 5	
1 2	
1 3	
3 4	
4 5	
5 3	
6 7	
1 2	
1 3	
2 3	
3 4	
4 5	
4 6	
5 6	
5 5	
1 2	
1 3	
2 3	
2 4	
3 5	
10 12	
1 2	
1 3	
2 3	
2 4	
4 5	
5 6	
6 7	
7 4	
3 8	
8 9	
9 10	
10 8	

Ghi chú

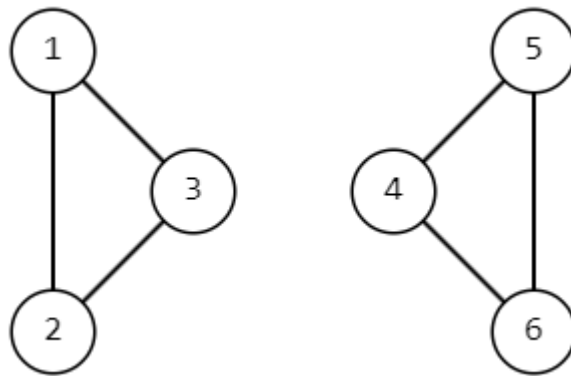
Trong tập dữ liệu đầu vào đầu tiên, chúng ta sẽ loại bỏ cạnh duy nhất (1, 2) và cặp đỉnh duy nhất (1, 2) sẽ trở nên không thể đến được từ nhau.

Trong tập dữ liệu đầu vào thứ hai, bất kỳ cạnh nào chúng ta loại bỏ, tất cả các đỉnh đều có thể đến được từ nhau.

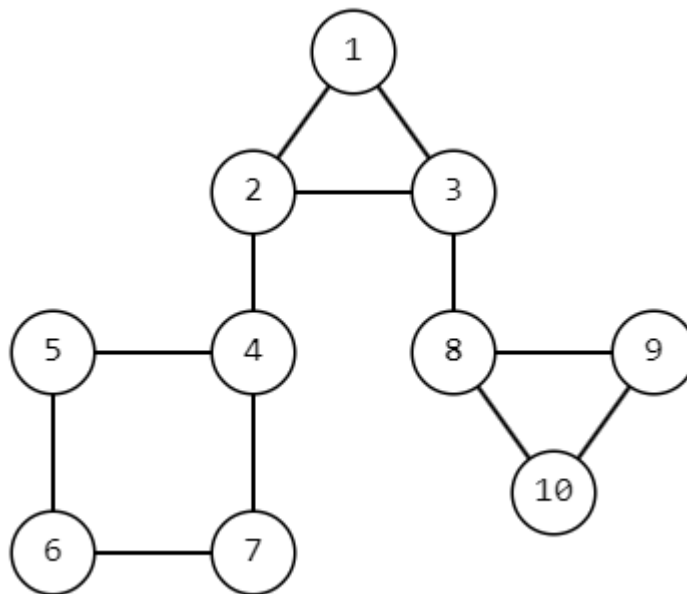
Trong tập dữ liệu đầu vào thứ tư, đồ thị ban đầu trông như thế này.



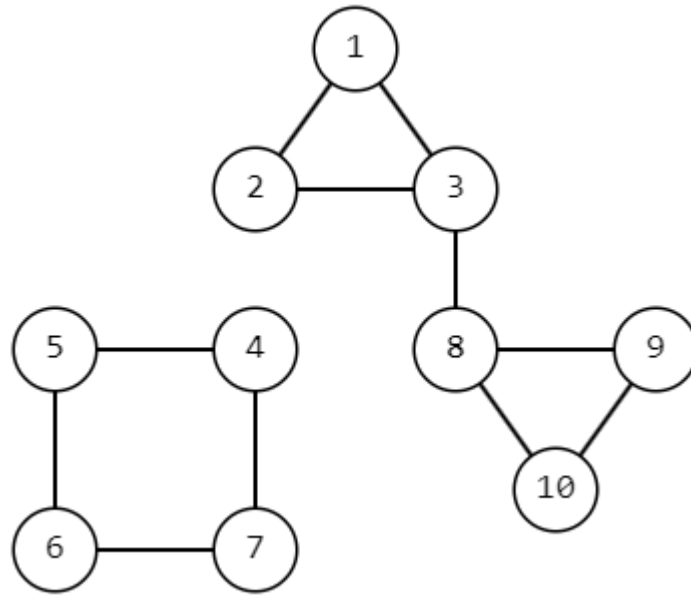
Chúng ta sẽ loại bỏ cạnh $(3, 4)$ và sau đó chỉ còn lại các cặp đỉnh có thể đến được là $(1, 2)$, $(1, 3)$, $(2, 3)$, $(4, 5)$, $(4, 6)$, $(5, 6)$.



Trong tập dữ liệu đầu vào thứ sáu, đồ thị ban đầu trông như thế này.



Sau khi loại bỏ cạnh $(2, 4)$, đồ thị sẽ trông như thế này. Do đó, sẽ có 21 cặp đỉnh có thể đến được.



Problem M. King Graffs Defense

Time Limit	2000 ms
Mem Limit	1572864 kB
Code Length Limit	50000 B
OS	Linux

King Graff, the ruler of the land of Feerie, has a problem - his nation is under attack! Luckily, he has an army at his disposal, composed of a whopping S soldiers (where $S = 2$).

Feerie consists of N ($1 \leq N \leq 100,000$) towns (numbered $1..N$), and M ($1 \leq M \leq 500,000$) roads. The i th road runs between distinct towns A_i and B_i , in both directions. No pair of towns is directly connected by more than one road, but every pair of towns is connected by at least one path of connected roads. King Graff would like to position his two soldiers in two different towns to prepare for the impending assault - however, since he's not much of a strategist, he'll choose the towns at complete random.

Graff's only real concern is with his enemies using a divide-and-conquer strategy. His soldiers will be susceptible to this type of attack if there exists any single road that, if blocked, will prevent them from reaching each other by any system of connected roads. As the royal computer scientist, your job is to determine the probability that King Graff will be defeated.

Input

First line: 2 integers, N and M

Next M lines: 2 integers, A_i and B_i , for $i = 1..M$

Output

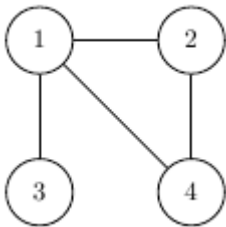
1 real number (rounded to 5 decimal places), the probability that the two towns chosen by Graff can be disconnected by the removal of any single road

Example

Input	Output
4 4 1 2 1 3 2 4 4 1	0.50000

Explanation of Sample:

The map of Feerie is illustrated below:



King Graff can make 6 possible choices as to where to place his soldiers, and three of those (the three with one of the soldiers being at town 3) result in defeat (if the road between towns 1 and 3 is destroyed). The probability of failure is then $3/6 = 0.5$.

Problem N. Police

Time Limit 1000 ms

Mem Limit 524288 kB

Để truy bắt tội phạm, cảnh sát xây dựng một hệ thống máy tính mới. Bản đồ khu vực bao gồm N thành phố và E đường nối 2 chiều. Các thành phố được đánh số từ 1 đến N .

Cảnh sát muốn bắt các tội phạm di chuyển từ thành phố này đến thành phố khác. Các điều tra viên, theo dõi bản đồ, phải xác định vị trí thiết lập trạm gác. Hệ thống máy tính mới phải trả lời được 2 loại truy vấn sau:

- 1. Đối với hai thành phố A, B và một đường nối giữa hai thành phố G_1, G_2 , hỏi tội phạm có thể di chuyển từ A đến B nếu đường nối này bị chặn (nghĩa là tên tội phạm không thể sử dụng con đường này) không?
- 2. Đối với 3 thành phố A, B, C , hỏi tội phạm có thể di chuyển từ A đến B nếu như toàn bộ thành phố C bị kiểm soát (nghĩa là tên tội phạm không thể đi vào thành phố này) không?

Input

- Dòng đầu tiên chứa 2 số nguyên N và E ($2 \leq N \leq 100000, 1 \leq E \leq 500000$), số thành phố và số đường nối.
- Mỗi dòng trong số E dòng tiếp theo chứa 2 số nguyên phân biệt thuộc phạm vi $[1, N]$ - cho biết nhãn của hai thành phố nối với nhau bởi một con đường. Giữa hai thành phố có nhiều nhất một đường nối.
- Dòng tiếp theo chứa số nguyên Q ($1 \leq Q \leq 300000$), số truy vấn được thử nghiệm trên hệ thống.
- Mỗi dòng trong Q dòng tiếp theo chứa 4 hoặc 5 số nguyên. Số đầu tiên cho biết loại truy vấn - 1 hoặc 2.
 - Nếu loại truy vấn là 1, tiếp theo trên cùng dòng là 4 số nguyên A, B, G_1, G_2 với ý nghĩa như đã mô tả. A khác B ; G_1, G_2 mô tả một con đường có sẵn.
 - Nếu loại truy vấn là 2, tiếp theo trên cùng dòng là 3 số nguyên A, B, C với ý nghĩa như đã mô tả. A, B, C đôi một khác nhau.

Dữ liệu được cho sao cho ban đầu luôn có cách di chuyển giữa hai thành phố bất kỳ.

Output

Gồm Q dòng, mỗi dòng chứa câu trả lời cho một truy vấn. Nếu câu trả lời là khẳng định, in ra "yes". Nếu câu trả lời là phủ định, in ra "no".

Input	Output
13 15	yes
1 2	yes
2 3	yes
3 5	no
2 4	yes
4 6	
2 6	
1 4	
1 7	
7 8	
7 9	
7 10	
8 11	
8 12	
9 12	
12 13	
5	
1 5 13 1 2	
1 6 2 1 4	
1 13 6 7 8	
2 13 6 7	
2 13 6 8	

Problem O. Dự báo thời tiết hôm nay mưa

Time Limit 1000 ms

Mem Limit 524288 kB

Hãng hàng không OlympAirways thực hiện các chuyến bay giữa n sân bay được đánh số từ 1 đến n . Hệ thống các chuyến bay được thiết lập sao cho giữa 2 sân bay bất kỳ được phục vụ bởi hãng luôn có một đường bay bao gồm một hoặc nhiều chuyến bay trực tiếp giữa hai sân bay. Mỗi chuyến bay thực hiện việc di chuyển giữa hai thành phố theo cả hai chiều.

Trung tâm điều khiển của hãng đưa ra khái niệm độ dính kết giữa cặp hai sân bay A và B được xác định như là số lượng các chuyến bay mà việc không thực hiện một trong số chúng (các chuyến bay khác vẫn thực hiện bình thường) dẫn đến không thể bay từ sân bay A đến sân bay B .

Một nghiên cứu cho biết rằng, trong điều kiện thời tiết xấu, tổng độ dính kết giữa các cặp sân bay phải đạt đến một giá trị nhất định thì hệ thống đường bay mới được gọi là an toàn.

Yêu cầu: Hãy giúp trung tâm điều khiển tính tổng độ dính kết giữa mọi cặp sân bay.

Input

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 100$)
- Dòng thứ hai chứa số nguyên m ($1 \leq m \leq 5000$) - số lượng các chuyến bay
- Mỗi dòng trong số m dòng tiếp theo chứa thông tin về một chuyến bay, bao gồm hai số nguyên dương trong khoảng từ 1 đến n : chỉ số của hai sân bay được nối bởi chuyến bay.

Output

In ra 1 số nguyên duy nhất là tổng độ dính kết giữa mọi cặp sân bay (A, B) (với $A < B$).

Input	Output
5 5 1 2 4 2 4 5 3 2 3 1	10

Problem P. Sài Gòn đau lòng quá

Time Limit 380 ms

Mem Limit 524288 kB

Đất nước Hạnh Phúc có N thành phố được nối với nhau bởi M đường nối hai chiều. Giữa hai thành phố bất kỳ chỉ có nhiều nhất một con đường.

Chính quyền nước này đưa ra một tiêu chí để đánh giá độ quan trọng của mỗi thành phố, theo đó độ quan trọng của một thành phố X được tính bằng số cặp thành phố A và B mà để di chuyển từ A đến B (và ngược lại) bắt buộc phải đi qua thành phố X .

Bạn hãy lập trình tính độ quan trọng trung bình của tất cả các thành phố.

Input

Dòng đầu tiên chứa hai số nguyên N, M ($1 \leq N \leq 20000, 0 \leq M \leq 200000$).

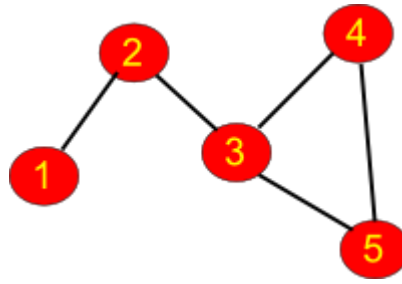
M dòng tiếp theo mỗi dòng chứa 2 số nguyên u, v ($1 \leq u, v \leq N$) mô tả một đường nối.

Output

Gồm một số thực duy nhất là độ quan trọng trung bình của các thành phố. Kết quả được tính là đúng nếu chênh lệch với đáp án không quá 10^{-2}

Input	Output
5 5 1 2 2 3 3 4 4 5 5 3	1.4000000000

Note



Giải thích: Độ quan trọng của các thành phố 1, 2, 3, 4, 5 lần lượt là 0, 3, 4, 0, 0. Độ quan trọng của thành phố 3 là 4 vì có 4 cặp thành phố mà khi di chuyển đến nhau cần đi qua thành phố 3: (1, 4), (1, 5), (2, 4), (2, 5).

Problem Q. bcc

Time Limit 380 ms

Mem Limit 524288 kB

Có N máy tính đánh số từ 1 đến N và M dây cáp mạng, giữa 2 máy tính có thể có một hoặc nhiều đường dây cáp mạng nối chúng, không có cáp mạng nối một máy với chính nó. Hai máy tính có thể truyền dữ liệu cho nhau nếu có đường cáp nối trực tiếp giữa chúng hoặc truyền qua một số máy trung gian.

Một tập S các máy tính được gọi là hệ thống an toàn nếu dù một máy tính bất kỳ bị tấn công (do sự tò mò của người dân :-(, cú thích truy cập và hack những trang cấm :-() thì trong số những máy tính còn lại, những máy tính thuộc tập S vẫn có thể truyền được dữ liệu cho nhau. Xác định số lượng lớn nhất có thể các máy tính của tập S

Input

- Dòng 1 chứa 2 số nguyên N, M ($1 \leq N \leq 30\,000, 0 \leq M \leq 100\,000$)
- M dòng tiếp theo ghi thông tin về các dây cáp mạng, gồm 2 chỉ số của 2 máy được dây đó nối trực tiếp

Output

Ghi một số nguyên duy nhất là số lượng máy tính lớn nhất tìm đc

Input	Output
8 10 1 2 2 3 3 1 1 4 4 5 5 1 1 6 6 7 7 8 8 1	4

Problem R. Thành phố đã quá chật

Time Limit 1000 ms

Mem Limit 524288 kB

Mạng giao thông của thành phố NB có n nút giao thông và m đoạn đường phố hai chiều nối các nút giao thông. Các nút giao thông được đánh số từ 1 đến n . Các đoạn đường phố được đánh số từ 1 đến m . Mạng giao thông của thành phố có tính chất sau đây:

- Giữa hai nút giao thông bất kỳ có không quá một đoạn đường phố nối chúng;
- Không có đoạn đường phố nào nối một nút giao thông với chính nó.

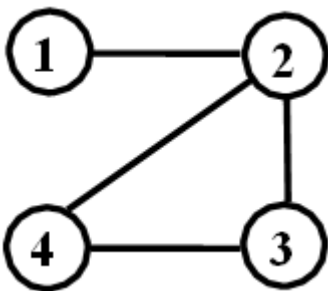
Vấn đề giao thông là thách thức với chính quyền thành phố từ nhiều năm. Với mong muốn đảm bảo việc đi lại thuận lợi hơn cho người dân, chính quyền thành phố quyết định tiến hành cải tổ mạng giao thông, trước hết nhằm đảm bảo có thể đi từ một nút giao thông bất kỳ đến tất cả các nút còn lại. (Lưu ý là mạng giao thông trước khi cải tổ có thể không đảm bảo yêu cầu này). Tuy nhiên, do hạn hẹp về nguồn kinh phí, trước mắt kế hoạch cải tổ chỉ có thể bao gồm 2 công việc:

- Loại bỏ một đoạn đường phố hiện có khỏi mạng giao thông;
- Xây dựng một đoạn đường chưa từng có trước đó nối hai nút giao thông khác nhau.

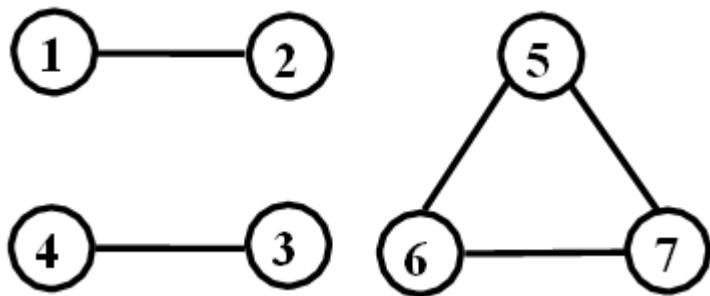
Đồng thời, sau khi thực hiện cải tổ, mạng giao thông phải đảm bảo có thể đi từ một nút giao thông bất kỳ đến tất cả các nút còn lại.

Yêu cầu: Giúp chính quyền thành phố xác định xem có bao nhiêu cách khác nhau để thực hiện kế hoạch cải tổ thỏa mãn các yêu cầu đặt ra. (Hai kế hoạch cải tổ là khác nhau nếu có ít nhất một trong hai đoạn đường được lựa chọn loại bỏ hay xây dựng mới là khác nhau.)

Hình vẽ minh họa: (VÍ DỤ 1)



Hình vẽ minh họa: (VÍ DỤ 2)



Input

- Dòng đầu tiên chứa hai số nguyên n và m ($m \leq 10^5, n \leq 2 \times 10^5$)
- Dòng thứ i trong số m dòng tiếp theo chứa hai số nguyên u_i và v_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i$) là chỉ số của hai nút giao thông được nối bởi đoạn đường i ($i = 1, 2, \dots, m$).

Hai số liên tiếp trên cùng dòng được ghi cách nhau bởi dấu cách.

Output

- Ghi ra một số nguyên là số cách thực hiện kế hoạch cải tổ mạng giao thông thỏa mãn các yêu cầu đặt ra.

Input	Output
4 4 1 2 2 3 2 4 3 4	8
Input	Output
7 5 1 2 3 4 5 6 5 7 6 7	0