

[Graph].Bài 1. Chuyển danh sách cạnh sang danh sách kề

Cho đồ thị vô hướng G = được biểu diễn dưới dạng danh sách cạnh. Hãy chuyển đổi đồ thị dưới dạng danh sách cạnh này sang danh sách kề tương ứng.

Input Format

Dòng đầu tiên là 2 số n và m , tương ứng với số lượng đỉnh và cạnh của đồ thị. Các đỉnh của đồ thị được đánh số từ 1 tới n . M dòng tiếp theo mỗi dòng chứa đỉnh u, v tương ứng với một cạnh của đồ thị.

Constraints

$1 \leq n \leq 1000$; $1 \leq m \leq n(n-1)/2$;

Output Format

In ra n dòng, trong đó dòng thứ i in ra danh sách kề của đỉnh i . Các đỉnh trong danh sách kề được liệt kê từ nhỏ tới lớn.

Sample Input 0

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

Sample Output 0

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

[Graph].Bài 2. Chuyển từ danh sách kề sang danh sách cạnh

Cho đồ thị vô hướng G = được biểu diễn dưới dạng danh sách kề. Hãy chuyển đổi đồ thị dưới dạng danh sách kề này sang danh sách cạnh tương ứng.

Input Format

Dòng đầu tiên là số nguyên dương n , tương ứng với số đỉnh của đồ thị. N dòng tiếp theo mỗi dòng chứa nhiều số nguyên, dòng thứ i tương ứng với các đỉnh trong danh sách kề của đỉnh i .

Constraints

$1 \leq n \leq 1000$;

Output Format

In ra các cạnh trong đồ thị trên từng dòng, các cạnh được liệt kê tăng dần. Chú ý mỗi cạnh chỉ liệt kê một lần.

Sample Input 0

```
5 4
```

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

Sample Output 0

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

[Graph].Bài 3. Chuyển danh sách cạnh sang danh sách kề (có hướng)

Cho đồ thị có hướng G = được biểu diễn dưới dạng danh sách cạnh. Hãy chuyển đổi đồ thị dưới dạng danh sách cạnh này sang danh sách kề tương ứng.

Input Format

Dòng đầu tiên là 2 số n và m , tương ứng với số lượng đỉnh và cạnh của đồ thị. Các đỉnh của đồ thị được đánh số từ 1 tới n . M dòng tiếp theo mỗi dòng chứa đỉnh u, v ($u \neq v$) tương ứng với một cạnh của đồ thị.

Constraints

$1 \leq n \leq 1000$; $1 \leq m \leq n \cdot (n-1)/2$;

Output Format

In ra n dòng, trong đó dòng thứ i in ra danh sách kề của đỉnh i . Các đỉnh trong danh sách kề được liệt kê từ nhỏ tới lớn.

Sample Input 0

```
5 10
4 2
4 1
2 5
3 4
1 5
2 3
5 1
2 1
1 4
4 5
```

Sample Output 0

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

[Graph]. Bài 4. Chuyển từ danh sách kề sang danh sách cạnh (có hướng).

Cho đồ thị có hướng G = được biểu diễn dưới dạng danh sách kề. Hãy chuyển đổi đồ thị dưới dạng danh sách kề này sang danh sách cạnh tương ứng.

Input Format

Dòng đầu tiên là số nguyên dương n , tương ứng với số đỉnh của đồ thị. N dòng tiếp theo mỗi dòng chứa nhiều số nguyên, dòng thứ i tương ứng với các đỉnh trong danh sách kề của đỉnh i .

Constraints

$1 \leq n \leq 1000$;

Output Format

In ra các cạnh trong đồ thị trên từng dòng, các cạnh được liệt kê tăng dần.

Sample Input 0

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

Sample Output 0

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

[Graph]. Bài 5. Ma trận kề sang danh sách kề, danh sách cạnh (vô hướng).

Cho ma trận kề $A[][]$ của đồ thị vô hướng G = chỉ bao gồm các số 0 hoặc 1. Hãy chuyển đổi đồ thị này thành danh sách cạnh và danh sách kề tương ứng.

Input Format

Dòng đầu tiên là n tương ứng với số lượng đỉnh của đồ thị. n dòng tiếp theo mỗi dòng gồm n số mô tả ma trận A

Constraints

$1 \leq n \leq 1000$

Output Format

Những dòng đầu tiên liệt kê danh sách cạnh theo thứ tự từ điển tăng dần, mỗi cạnh chỉ liệt kê một lần và liệt kê đỉnh bắt đầu là đỉnh lớn hơn. Sau đó cách ra 1 dòng và liệt kê danh sách kề, trong đó các đỉnh trong danh sách kề được in ra theo thứ tự tăng dần. Xem output mẫu để rõ hơn.

Sample Input 0

```
5
0 1 1 1 0
1 0 0 1 1
1 0 0 1 1
1 1 1 0 1
0 1 1 1 0
```

Sample Output 0

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

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

[Graph]. Bài 6. Ma trận kề sang danh sách kề, danh sách cạnh (có hướng).

Cho ma trận kề $A[][]$ của đồ thị có hướng $G = (V, E)$ chỉ bao gồm các số 0 hoặc 1. Hãy chuyển đổi đồ thị này thành danh sách cạnh và danh sách kề tương ứng.

Input Format

Dòng đầu tiên là n tương ứng với số lượng đỉnh của đồ thị. n dòng tiếp theo mỗi dòng gồm n số mô tả ma trận A

Constraints

$1 \leq n \leq 1000$

Output Format

Những dòng đầu tiên liệt kê danh sách cạnh theo thứ tự từ điển tăng dần. Sau đó cách ra 1 dòng và liệt kê danh sách kề, trong đó các đỉnh trong danh sách kề được in ra theo thứ tự tăng dần. Xem output mẫu để rõ hơn.

Sample Input 0

```
5
0 0 0 1 0
1 0 1 1 1
1 0 0 1 1
1 0 0 0 0
0 1 0 1 0
```

Sample Output 0

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

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

[Graph]. Bài 7. Danh sách cạnh sang ma trận kề (vô hướng).

Cho đồ thị vô hướng G = được biểu diễn dưới dạng danh sách cạnh. Hãy chuyển đổi đồ thị dưới dạng danh sách cạnh này sang ma trận kề tương ứng.

Input Format

Dòng đầu tiên là 2 số n và m , tương ứng với số lượng đỉnh và cạnh của đồ thị. Các đỉnh của đồ thị được đánh số từ 1 tới n . m dòng tiếp theo mỗi dòng chứa đỉnh u, v tương ứng với một cạnh của đồ thị.

Constraints

$1 \leq n \leq 1000$; $1 \leq m \leq n(n-1)/2$;

Output Format

In ra ma trận kề theo mẫu.

Sample Input 0

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

Sample Output 0

```
0 0 1 0 0
0 0 0 1 0
1 0 0 1 0
0 1 1 0 1
0 0 0 1 0
```

[Graph]. Bài 8. Ma trận trọng số sang danh sách cạnh

Cho đồ thị vô hướng G = được biểu diễn dưới ma trận trọng số $A[i][j]$, trong đó $A[i][j]$ khác không có nghĩa có cạnh nối giữa 2 đỉnh i, j . Hãy chuyển đồ thị đã cho thành danh sách cạnh tương ứng.

Input Format

Dòng đầu tiên là n , tương ứng là số lượng đỉnh của đồ thị. N dòng tiếp theo là ma trận $A[i][j]$.

Constraints

$1 \leq n \leq 1000$; $1 \leq A[i][j] \leq 10^6$;

Output Format

In ra danh sách cạnh và trọng số theo thứ tự đỉnh tăng dần.

Sample Input 0

```
5
0 0 9 4 1
0 0 1 0 6
9 1 0 3 1
4 0 3 0 8
1 6 1 8 0
```

Sample Output 0

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

[Graph]. Bài 9. DFS trên đồ thị vô hướng

Cho đồ thị vô hướng G = được biểu diễn dưới dạng danh sách cạnh. Hãy thực hiện in ra danh sách các đỉnh được duyệt theo thuật toán DFS(s).

Input Format

Dòng đầu tiên là 2 số n và m và s , tương ứng với số lượng đỉnh, cạnh của đồ thị và đỉnh bắt đầu duyệt. Các đỉnh của đồ thị được đánh số từ 1 tới n . m dòng tiếp theo mỗi dòng chứa đỉnh u, v ($u \neq v$) tương ứng với một cạnh của đồ thị.

Constraints

$1 \leq s \leq n \leq 1000$; $1 \leq m \leq n \cdot (n-1)/2$;

Output Format

In ra các đỉnh được duyệt theo thuật toán DFS(s). Chú ý trong quá trình mở rộng các đỉnh của thuật toán DFS luôn lựa chọn duyệt các đỉnh có thứ tự nhỏ hơn trước.

Sample Input 0

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

Sample Output 0

```
3 2 5 1 4
```

[Graph]. Bài 10. DFS trên đồ thị có hướng

Cho đồ thị có hướng $G = (V, E)$ được biểu diễn dưới dạng danh sách cạnh. Hãy thực hiện in ra danh sách các đỉnh được duyệt theo thuật toán DFS(s).

Input Format

Dòng đầu tiên là 2 số n và m và s , tương ứng với số lượng đỉnh, cạnh của đồ thị và đỉnh bắt đầu duyệt. Các đỉnh của đồ thị được đánh số từ 1 tới n . m dòng tiếp theo mỗi dòng chứa đỉnh u, v ($u \neq v$) tương ứng với một cạnh của đồ thị.

Constraints

$1 \leq s \leq n \leq 1000$; $1 \leq m \leq n \cdot (n-1)/2$;

Output Format

In ra các đỉnh được duyệt theo thuật toán DFS(s). Chú ý trong quá trình mở rộng các đỉnh của thuật toán DFS luôn lựa chọn duyệt các đỉnh có thứ tự nhỏ hơn trước.

Sample Input 0

```
5 10 4
5 1
4 5
4 2
4 3
1 5
3 5
5 3
3 1
1 4
5 2
```

Sample Output 0

```
4 2 3 1 5
```

[Graph]. Bài 11. BFS trên đồ thị vô hướng.

Cho đồ thị vô hướng $G = (V, E)$ được biểu diễn dưới dạng danh sách cạnh. Hãy thực hiện in ra danh sách các đỉnh được duyệt theo thuật toán BFS(s).

Input Format

Dòng đầu tiên là 2 số n và m và s , tương ứng với số lượng đỉnh, cạnh của đồ thị và đỉnh bắt đầu duyệt. Các đỉnh của đồ thị được đánh số từ 1 tới n . m dòng tiếp theo mỗi dòng chứa đỉnh u, v ($u \neq v$) tương ứng với một cạnh của đồ thị.

Constraints

$1 \leq s \leq n \leq 1000$; $1 \leq m \leq n*(n-1)/2$;

Output Format

In ra các đỉnh được duyệt theo thuật toán BFS(s). Chú ý trong quá trình mở rộng các đỉnh của thuật toán BFS luôn lựa chọn duyệt các đỉnh có thứ tự nhỏ hơn trước.

Sample Input 0

```
5 7 5
4 1
5 3
4 3
5 1
3 2
4 2
2 1
```

Sample Output 0

```
5 1 3 2 4
```

[Graph]. Bài 12. BFS trên đồ thị có hướng

Cho đồ thị có hướng $G = (V, E)$ được biểu diễn dưới dạng danh sách cạnh. Hãy thực hiện in ra danh sách các đỉnh được duyệt theo thuật toán BFS(s).

Input Format

Dòng đầu tiên là 2 số n và m và s , tương ứng với số lượng đỉnh, cạnh của đồ thị và đỉnh bắt đầu duyệt. Các đỉnh của đồ thị được đánh số từ 1 tới n . m dòng tiếp theo mỗi dòng chứa đỉnh u, v ($u \neq v$) tương ứng với một cạnh của đồ thị.

Constraints

$1 \leq s \leq n \leq 1000$; $1 \leq m \leq n*(n-1)/2$;

Output Format

In ra các đỉnh được duyệt theo thuật toán BFS(s). Chú ý trong quá trình mở rộng các đỉnh của thuật toán BFS luôn lựa chọn duyệt các đỉnh có thứ tự nhỏ hơn trước.

Sample Input 0

```
5 9 4
5 1
4 5
2 5
4 3
1 4
2 4
```



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

Sample Output 0

```
4 3 5 1 2
```

[Graph]. Bài 13. Đếm số thành phần liên thông

Cho đồ thị vô hướng $G = (V, E)$ được biểu diễn dưới dạng danh sách cạnh. Hãy đếm số thành phần liên thông của đồ thị.

Input Format

Dòng đầu tiên là 2 số n và m , tương ứng với số lượng đỉnh, cạnh của đồ thị. Các đỉnh của đồ thị được đánh số từ 1 tới n . m dòng tiếp theo mỗi dòng chứa đỉnh u, v ($u \neq v$) tương ứng với một cạnh của đồ thị.

Constraints

$1 \leq n \leq 1000; 1 \leq m \leq n(n-1)/2;$

Output Format

In ra số thành phần liên thông của đồ thị

Sample Input 0

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

Sample Output 0

```
2
```

Sample Input 1

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

Sample Output 1

```
3
```

[Graph]. Bài 14. Đường đi trên đồ thị vô hướng bằng DFS

Cho đồ thị vô hướng $G = (V, E)$ được biểu diễn dưới dạng danh sách cạnh. Hãy tìm đường đi theo thuật toán DFS từ đỉnh s tới đỉnh t . Trong quá trình mở rộng của thuật toán DFS, luôn ưu tiên mở rộng đỉnh có số thứ tự nhỏ hơn.

Input Format

Dòng đầu tiên là 4 số n, m, s, t , tương ứng với số lượng đỉnh, cạnh của đồ thị, đỉnh bắt đầu và đỉnh kết thúc. Các đỉnh của đồ thị được đánh số từ 1 tới n . m dòng tiếp theo mỗi dòng chứa đỉnh u, v ($u \neq v$) tương ứng với một cạnh của đồ thị.

Constraints

$1 \leq s, t \leq n \leq 1000$; $1 \leq m \leq n*(n-1)/2$;

Output Format

In ra đường đi từ s tới t nếu có đường đi, trường hợp không tồn tại đường đi thì in ra -1.

Sample Input 0

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

Sample Output 0

```
4 2 1 3
```

[Graph]. Bài 15. Đường đi trên đồ thị vô hướng bằng BFS

Cho đồ thị vô hướng $G = (V, E)$ được biểu diễn dưới dạng danh sách cạnh. Hãy tìm đường đi theo thuật toán BFS từ đỉnh s tới đỉnh t. Trong quá trình mở rộng của thuật toán BFS, luôn ưu tiên mở rộng đỉnh có số thứ tự nhỏ hơn.

Input Format

Dòng đầu tiên là 4 số n, m, s, t, tương ứng với số lượng đỉnh, cạnh của đồ thị, đỉnh bắt đầu và đỉnh kết thúc. Các đỉnh của đồ thị được đánh số từ 1 tới n. m dòng tiếp theo mỗi dòng chứa đỉnh u, v ($u \neq v$) tương ứng với một cạnh của đồ thị.

Constraints

$1 \leq s, t \leq n \leq 1000$; $1 \leq m \leq n*(n-1)/2$;

Output Format

In ra đường đi từ s tới t nếu có đường đi, trường hợp không tồn tại đường đi thì in ra -1.

Sample Input 0

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

Sample Output 0

```
2 4
```

[Graph]. Bài 16. Đường đi trên đồ thị có hướng bằng DFS

Cho đồ thị có hướng $G = (V, E)$ được biểu diễn dưới dạng danh sách cạnh. Hãy tìm đường đi theo thuật toán DFS từ đỉnh s tới đỉnh t. Trong quá trình mở rộng của thuật toán DFS, luôn ưu tiên mở rộng đỉnh có số thứ tự nhỏ hơn.

Input Format

Dòng đầu tiên là 4 số n, m, s, t , tương ứng với số lượng đỉnh, cạnh của đồ thị, đỉnh bắt đầu và đỉnh kết thúc. Các đỉnh của đồ thị được đánh số từ 1 tới n . m dòng tiếp theo mỗi dòng chứa đỉnh u, v ($u \neq v$) tương ứng với một cạnh của đồ thị.

Constraints

$1 \leq s, t \leq n \leq 1000; 1 \leq m \leq n*(n-1)/2;$

Output Format

In ra đường đi từ s tới t nếu có đường đi, trường hợp không tồn tại đường đi thì in ra -1.

Sample Input 0

```
5 8 1 2
5 3
5 2
4 1
4 3
2 5
3 4
1 5
3 1
```

Sample Output 0

```
1 5 2
```

[Graph]. Bài 17. Đường đi trên đồ thị có hướng bằng BFS

Cho đồ thị có hướng $G = (V, E)$ được biểu diễn dưới dạng danh sách cạnh. Hãy tìm đường đi theo thuật toán BFS từ đỉnh s tới đỉnh t . Trong quá trình mở rộng của thuật toán BFS, luôn ưu tiên mở rộng đỉnh có số thứ tự nhỏ hơn.

Input Format

Dòng đầu tiên là 4 số n, m, s, t , tương ứng với số lượng đỉnh, cạnh của đồ thị, đỉnh bắt đầu và đỉnh kết thúc. Các đỉnh của đồ thị được đánh số từ 1 tới n . m dòng tiếp theo mỗi dòng chứa đỉnh u, v ($u \neq v$) tương ứng với một cạnh của đồ thị.

Constraints

$1 \leq s, t \leq n \leq 1000; 1 \leq m \leq n*(n-1)/2;$

Output Format

In ra đường đi từ s tới t nếu có đường đi, trường hợp không tồn tại đường đi thì in ra -1.

Sample Input 0

```
5 10 2 3
5 1
4 5
3 5
4 3
2 1
3 2
5 3
```

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

Sample Output 0

```
2 1 3
```

[Graph]. Bài 18. Kiểm tra đường đi

Cho đồ thị vô hướng $G = (V, E)$ được biểu diễn dưới dạng danh sách cạnh. Có Q truy vấn, mỗi truy vấn yêu cầu trả lời câu hỏi giữa 2 đỉnh s và t có tồn tại đường đi tới nhau hay không ?

Input Format

Dòng đầu tiên là 2 số n, m , tương ứng với số lượng đỉnh, cạnh của đồ thị. Các đỉnh của đồ thị được đánh số từ 1 tới n . m dòng tiếp theo mỗi dòng chứa đỉnh u, v ($u \neq v$) tương ứng với một cạnh của đồ thị. Dòng tiếp theo là Q , Q dòng tiếp theo mỗi dòng chứa 2 đỉnh s, t cần truy vấn.

Constraints

$1 \leq s, t \leq n \leq 1000$; $1 \leq m \leq n \cdot (n-1)/2$; $1 \leq Q \leq 10000$;

Output Format

Đối với mỗi truy vấn in ra 1 nếu có đường đi giữa s và t , ngược lại in ra -1.

Sample Input 0

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

Sample Output 0

```
1
-1
1
```

[Graph]. Bài 19. Đỉnh trụ

Cho đồ thị vô hướng $G = (V, E)$ được biểu diễn dưới dạng danh sách cạnh. Hãy đếm số lượng đỉnh trụ của đồ thị.

Input Format

Dòng đầu tiên là 2 số n, m , tương ứng với số lượng đỉnh, cạnh của đồ thị. Các đỉnh của đồ thị được đánh số từ 1 tới n . m dòng tiếp theo mỗi dòng chứa đỉnh u, v ($u \neq v$) tương ứng với một cạnh của đồ thị.

Constraints

$1 \leq n \leq 1000; 1 \leq m \leq n*(n-1)/2;$

Output Format

In ra số lượng đỉnh trụ của đồ thị

Sample Input 0

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

Sample Output 0

```
3
```

Explanation 0

3 đỉnh trụ của đồ thị là : 2, 5, 10

[Graph]. Bài 20. Cạnh cầu

Cho đồ thị vô hướng $G = (V, E)$ được biểu diễn dưới dạng danh sách cạnh. Hãy đếm số lượng cạnh cầu của đồ thị.

Input Format

Dòng đầu tiên là 2 số n, m , tương ứng với số lượng đỉnh, cạnh của đồ thị. Các đỉnh của đồ thị được đánh số từ 1 tới n . m dòng tiếp theo mỗi dòng chứa đỉnh u, v ($u \neq v$) tương ứng với một cạnh của đồ thị. Giữa 2 đỉnh bất kỳ chỉ tồn tại nhiều nhất một cạnh nối.

Constraints

$1 \leq n \leq 1000; 1 \leq m \leq n*(n-1)/2;$

Output Format

In ra số lượng cạnh cầu của đồ thị

Sample Input 0

```
10 6
10 1
5 2
10 5
10 6
5 3
10 4
```

Sample Output 0

```
6
```

Sample Input 1

```
10 8
10 9
10 8
```

```
10 3
10 4
5 3
10 1
5 1
5 4
```

Sample Output 1

```
2
```

[Graph]. Bài 21. Kiểm tra chu trình trên đồ thị vô hướng

Cho đồ thị vô hướng $G = (V, E)$ được biểu diễn dưới dạng danh sách cạnh. Hãy xác định xem đồ thị có tồn tại chu trình hay không? Bài này các bạn cài đặt bằng 3 phương pháp : DFS, BFS và DSU.

Input Format

Dòng đầu tiên là 2 số n, m , tương ứng với số lượng đỉnh, cạnh của đồ thị. Các đỉnh của đồ thị được đánh số từ 1 tới n . m dòng tiếp theo mỗi dòng chứa đỉnh u, v ($u \neq v$) tương ứng với một cạnh của đồ thị.

Constraints

$1 \leq n \leq 1000; 1 \leq m \leq n*(n-1)/2;$

Output Format

In ra 1 nếu đồ thị tồn tại chu trình, ngược lại in ra 0.

Sample Input 0

```
10 11
10 5
10 4
10 1
10 3
5 2
5 4
10 8
5 3
5 1
10 6
10 9
```

Sample Output 0

```
1
```

[Graph]. Bài 22. Kiểm tra chu trình trên đồ thị có hướng

Cho đồ thị có hướng $G = (V, E)$ được biểu diễn dưới dạng danh sách cạnh. Hãy xác định xem đồ thị có tồn tại chu trình hay không? Bài này các bạn cài đặt bằng 3 phương pháp : DFS, BFS và DSU.

Input Format

Dòng đầu tiên là 2 số n, m , tương ứng với số lượng đỉnh, cạnh của đồ thị. Các đỉnh của đồ thị được đánh số từ 1 tới n . m dòng tiếp theo mỗi dòng chứa đỉnh u, v ($u \neq v$) tương ứng với một cạnh của đồ thị.

Constraints

$1 \leq n \leq 1000; 1 \leq m \leq n*(n-1)/2;$

Output Format

In ra 1 nếu đồ thị tồn tại chu trình, ngược lại in ra 0.

Sample Input 0

```
10 13
6 5
6 3
6 1
6 2
9 5
3 9
9 7
3 7
3 2
6 7
6 9
3 5
6 8
```

Sample Output 0

```
0
```

Sample Input 1

```
10 16
9 10
6 4
6 1
6 3
9 2
3 9
9 4
3 8
3 5
6 7
6 10
9 6
6 8
3 7
6 5
9 7
```

Sample Output 1

```
1
```

[Graph]. Bài 23. Đồ thị liên thông mạnh

Cho đồ thị có hướng $G = (V, E)$ được biểu diễn dưới dạng danh sách cạnh. Hãy kiểm tra đồ thị có liên thông mạnh hay không? Bài này các bạn thử cài đặt bằng 3 cách là : brute force, thuật toán Tarjan, thuật toán Kosaraju.

Input Format

Dòng đầu tiên là 2 số n, m , tương ứng với số lượng đỉnh, cạnh của đồ thị. Các đỉnh của đồ thị được đánh số từ 1 tới n . m dòng tiếp theo mỗi dòng chứa đỉnh u, v ($u \neq v$) tương ứng với một cạnh của đồ thị.

Constraints

$1 \leq n \leq 1000$; $1 \leq m \leq n \cdot (n-1)/2$;

Output Format

In ra 1 nếu đồ thị liên thông mạnh, ngược lại in ra 0.

Sample Input 0

```
6 14
3 4
3 1
2 5
2 6
4 6
2 3
5 6
1 5
1 2
3 5
4 5
6 3
4 2
1 4
```

Sample Output 0

```
1
```

[Graph]. Bài 24. Kiểm tra cây

Một đồ thị N đỉnh là một cây, nếu như nó có đúng $N-1$ cạnh và giữa 2 đỉnh bất kì, chỉ tồn tại duy nhất 1 đường đi giữa chúng. Cho một đồ thị vô hướng N đỉnh và M cạnh, hãy kiểm tra đồ thị đã cho có phải là một cây hay không?

Input Format

Dòng đầu tiên là số n, m tương ứng với số lượng đỉnh, cạnh của đồ thị. Các đỉnh của đồ thị được đánh số từ 1 tới n . m dòng tiếp theo mỗi dòng chứa đỉnh u, v ($u \neq v$) tương ứng với một cạnh của đồ thị.

Constraints

$1 \leq n \leq 1000; 1 \leq m \leq n*(n-1)/2;$

Output Format

In ra 1 nếu đồ thị đã cho là cây, ngược lại in ra 0.

Sample Input 0

```
6 5
4 3
2 1
6 5
5 4
3 2
```

Sample Output 0

```
1
```

Sample Input 1

```
6 4
6 3
3 2
4 2
6 4
```

Sample Output 1

```
0
```

[Graph]. Bài 25. Đếm số phòng

Bạn được đưa cho một bản đồ của một tòa nhà và nhiệm vụ của bạn là đếm số lượng phòng của tòa nhà đó. Kích thước của bản đồ là $n \times m$ hình vuông và mỗi hình vuông là sàn hoặc tường. Bạn có thể đi bộ sang trái, phải, lên và xuống qua các ô sàn.

Input Format

Dòng nhập đầu tiên có hai số nguyên n và m : chiều cao và chiều rộng của bản đồ. Khi đó có n dòng gồm m ký tự mô tả bản đồ. Mỗi ký tự là một trong hai ký tự '.' hoặc '#' tương ứng với sàn nhà và tường.

Constraints

$1 \leq n, m \leq 1000$

Output Format

In ra số nguyên duy nhất là số phòng của tòa nhà

Sample Input 0

```
6 6
..###.
...#..
#...#.
.##.#.
#####
```

```
. # . # # .
```

Sample Output 0

```
7
```

Sample Input 1

```
6 6
```

```
### # . #
```

```
. # . . # #
```

```
. # . . # #
```

```
### # . .
```

```
# . . . # #
```

```
. # # . . #
```

Sample Output 1

```
6
```

[Graph]. Bài 26. Xây dựng đường đi

Ở 28techland có n thành phố và m đường giữa chúng. Mục tiêu là xây dựng những con đường mới để có một tuyến đường giữa hai thành phố bất kỳ. Nhiệm vụ của bạn là tìm ra số lượng đường tối thiểu cần thiết, đồng thời xác định những con đường nào nên được xây dựng.

Input Format

Dòng đầu tiên có hai số nguyên n và m : số thành phố và đường. Các thành phố được đánh số từ 1 tới n . Sau đó, có m dòng mô tả các con đường. Mỗi dòng có hai số nguyên a và b : có một con đường giữa các thành phố đó, các con đường này là đường 2 chiều. Một con đường luôn nối hai thành phố khác nhau và có nhiều nhất một con đường giữa hai thành phố bất kỳ.

Constraints

$1 \leq n \leq 1000$; $0 \leq m \leq n(n-1)/2$; $1 \leq a, b \leq n$;

Output Format

Dòng đầu tiên in ra K là số con đường cần xây dựng. K dòng tiếp theo in ra các con đường được xây dựng, do có nhiều cách xây dựng đường nên bạn sẽ lựa chọn ra các con đường có thứ tự từ điển nhỏ nhất để xây dựng nhưng với một điều kiện đó là, mỗi thành phố được lựa chọn là điểm xuất phát của con đường không quá 1 lần.

Sample Input 0

```
6 3
```

```
4 3
```

```
4 1
```

```
4 2
```

Sample Output 0

```
2
```

```
1 5
```

```
5 6
```

Explanation 0

Có thể có nhiều cách xây dựng đường đi ví dụ : xây dựng đường đi giữa thành phố 2 và 5, 1 và 6 sẽ kết nối mọi thành phố. Tuy nhiên do yêu cầu của đầu bài, bạn sẽ phải xây dựng 2 con đường là 1->5, 5->6. Không thể xây dựng 1->5, 1->6 vì như vậy thành phố 1 sẽ 2 lần là điểm khởi đầu của con đường.

[Graph]. Bài 27. Nhóm bạn

Trong một lớp học luôn tồn tại những nhóm học sinh, các học sinh trong nhóm này sẽ thân thiết với nhau hơn. Giả sử trong 1 nhóm, nếu bạn a chơi thân vs bạn b và bạn b chơi thân với bạn c, thì bạn a và bạn c sẽ chơi thân với nhau. Hiện nay Tèo là lớp trưởng lớp 10A1, vì không thích việc chia bè kéo cánh trong lớp nên Tèo quyết định tìm ra nhóm có số bạn chơi thân với nhau nhất trong nhóm để đưa các thành viên của các nhóm khác vào trong nhóm có số lượng thành viên cao nhất này. Bạn hãy giúp Tèo tìm ra số lượng thành viên lớn nhất của 1 nhóm chơi thân bất kỳ trong lớp nhé !

Input Format

Dòng đầu tiên là n và m, tương ứng với số lượng sinh viên trong lớp và số lượng tình bạn giữa các bạn trong lớp. Các sinh viên được đánh số từ 1 tới n. M dòng tiếp theo mỗi dòng gồm 2 số a, b thể hiện tình bạn thân thiết giữa bạn a và bạn b, a và b khác nhau.

Constraints

$1 \leq n \leq 1000$; $1 \leq m \leq n*(n-1)/2$; $1 \leq a, b \leq n$;

Output Format

In ra nhóm có số lượng bạn lớn nhất

Sample Input 0

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

Sample Output 0

```
7
```

[Graph]. Bài 28. Đi tìm nhóm trưởng

Trong một lớp học luôn tồn tại những nhóm học sinh, các học sinh trong nhóm này sẽ thân thiết với nhau hơn. Giả sử trong 1 nhóm, nếu bạn a chơi thân vs bạn b và bạn b chơi thân với bạn c, thì bạn a và bạn c sẽ chơi thân với nhau. Hiện nay Tèo là lớp trưởng lớp 10A1, vì không thích việc chia bè kéo cánh trong lớp nên Tèo quyết định tìm các bạn nhóm trưởng của các nhóm để bàn việc hợp nhất các nhóm. Tèo biết rằng trong các nhóm nhỏ, nhóm trưởng là

người có nhiều quan hệ thân thiết với các thành viên khác trong nhóm nhất và là người có số thứ tự nhỏ nhất. Nếu một bạn nào đó không chơi với ai cả thì bạn đó là nhóm trưởng của chính mình. Bạn hãy giúp tèo in ra thứ tự của các trưởng nhóm theo thứ tự từ nhỏ đến lớn nhé.

Input Format

Dòng đầu tiên là n và m , tương ứng với số lượng sinh viên trong lớp và số lượng tình bạn giữa các bạn trong lớp. Các sinh viên được đánh số từ 1 tới n . M dòng tiếp theo mỗi dòng gồm 2 số a, b thể hiện tình bạn thân thiết giữa bạn a và bạn b , a và b khác nhau.

Constraints

$1 \leq n \leq 1000$; $1 \leq m \leq n*(n-1)/2$; $1 \leq a, b \leq n$;

Output Format

In ra thứ tự của các trưởng nhóm theo thứ tự từ nhỏ đến lớn nhé.

Sample Input 0

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

Sample Output 0

```
4 7 9 10
```

[Graph]. Bài 29. Đường đi trong mê cung

Cho mê cung A có n hàng, n cột. Một con chuột xuất phát từ ô (s, t) và di chuyển tới ô (u, v) . Nhiệm vụ của bạn là xác định số bước đi tối thiểu để con chuột con thể di chuyển từ ô (s, t) tới ô (u, v) hoặc xác định rằng không thể tìm được đường tới ô (u, v) . Mê cung A chỉ bao gồm các số 1 hoặc 0. Trong đó 1 đại diện cho đường đi và 0 đại diện cho vật cản, con chuột chỉ có thể di chuyển từ ô hiện tại sang các ô chung đỉnh với nó và có đường đi.

Input Format

Dòng đầu tiên là số nguyên dương N . Dòng thứ 2 gồm 4 số nguyên s, t, u, v . Dữ liệu đảm bảo ô (s, t) có giá trị 1. N dòng tiếp theo, mỗi dòng gồm N số nguyên mô tả mê cung A .

Constraints

$2 \leq n \leq 1000$; $0 \leq A[i][j] \leq 1$;

Output Format

In ra số bước đi tối thiểu của con chuột hoặc in ra -1 nếu con chuột không thể đi đến nơi.

Sample Input 0

```
10
9 6 9 3
1 1 0 1 1 1 1 0 0 1
```

```

0 0 1 0 0 1 0 1 0 1
1 1 1 1 0 0 0 1 1 0
1 0 0 0 1 0 0 0 1 1
1 0 1 0 0 1 0 1 1 0
0 0 1 1 0 1 0 0 0 0
1 1 0 1 0 1 1 0 0 0
0 0 0 1 1 0 1 1 0 1
1 0 1 0 0 1 0 0 1 1
0 1 1 1 1 0 1 0 1 1

```

Sample Output 0

```
3
```

[Graph]. Bài 30. Đường đi của quân mã

Cho bàn cờ A cỡ $n \times n$, một con mã ở ô (s, t) muốn di chuyển sang ô (u, v) . Hãy tìm số bước tối thiểu để con mã có thể đến ô (u, v) hoặc xác định rằng con mã không thể đi tới nơi. Bàn cờ A được mô tả bằng các giá trị 0, 1, trong đó ô có giá trị 1 thì con mã có thể tự do di chuyển đến ô này, ngược lại nếu giá trị tại một ô của bàn cờ là 0 thì con mã không thể di chuyển đến ô này.

Input Format

Dòng đầu tiên là số nguyên dương N . Dòng thứ 2 gồm 4 số nguyên s, t, u, v . Dữ liệu đảm bảo ô (s, t) có giá trị 1. N dòng tiếp theo, mỗi dòng gồm N số nguyên mô tả bàn cờ.

Constraints

$2 \leq n \leq 1000$; $0 \leq A[i][j] \leq 1$;

Output Format

In ra số bước đi tối thiểu của con mã hoặc in ra -1 nếu con mã không thể đi đến nơi.

Sample Input 0

```

10
6 7 8 9
0 1 1 0 1 0 1 0 1 0
1 1 0 0 1 1 0 0 1 0
1 1 0 1 1 0 0 1 0 0
0 1 1 1 1 1 1 1 1 0
0 0 1 0 0 1 1 1 1 1
1 1 0 1 1 0 1 0 0 0
1 0 0 0 0 1 1 0 1 1
1 1 1 1 1 1 1 0 1 1
1 0 0 1 1 1 0 1 1 1
1 0 0 1 1 0 0 1 0 1

```

Sample Output 0

```
4
```

[Graph]. Bài 31. Tìm cặp

Bạn được cung cấp một cây gốc với các nút và nút 1 là gốc. Có một đường đi duy nhất giữa hai nút bất kỳ. Ở đây, $d(i, j)$ được định nghĩa là số cạnh trong một đường đi ngắn nhất giữa 2 nút i và j . Nhiệm vụ của bạn phải tìm số lượng cặp i, j sao cho $d(i, j) = d(1, i) - d(1, j)$.

Input Format

Dòng đầu tiên là N là số lượng nút của cây. $N - 1$ dòng tiếp theo là các cạnh của cây.

Constraints

$1 \leq N \leq 1000$;

Output Format

In ra số lượng cặp (i, j) thỏa mãn yêu cầu của đầu bài.

Sample Input 0

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

Sample Output 0

```
51
```

Sample Input 1

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

Sample Output 1

```
10
```

Explanation 1

Các cặp thỏa mãn : $(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 3), (3, 4), (4, 4)$.

[Graph]. Bài 32. Diện tích

Bạn được cung cấp một lưới 2D. Dấu '#' đại diện cho chướng ngại vật và dấu '.' đại diện cho không gian trống. Bạn cần tìm các khu vực của các thành phần bị ngắt kết nối. Các ô $(i + 1, j)$, $(i, j + 1)$, $(i - 1, j)$, $(i, j - 1)$ là các ô liền kề với ô (i, j) .

Input Format

Dòng đầu tiên là n và m , tương ứng với số dòng và số cột của lưới 2D. N dòng tiếp theo, mỗi dòng là m ký tự '#' hoặc '.'.

Constraints

$1 \leq N, M \leq 1000$;

Output Format

In ra diện tích các vùng không gian trống liền kề trên 1 dòng.

Sample Input 0

```
10 10
###.#.#..#
.#####...
###..##.#.
.##...###.
##....#..#
.#...###...
..#####.##.
...##.....
..#...####
.#...##.##.
```

Sample Output 0

```
1 1 8 1 11 1 17 9 1 1
```

[Graph]. Bài 33. Mạng xã hội(SPOJ)

Tý đang xây dựng một mạng xã hội và mời các bạn của mình dùng thử. Bạn của bạn cũng là bạn. Vì vậy, Tý muốn mạng xã hội của mình là hoàn hảo, tức với mọi bộ ba X, Y, Z, nếu X kết bạn với Y, Y kết bạn với Z thì X và Z cũng phải là bạn bè của nhau trên mạng xã hội. Các bạn hãy xác định xem mạng xã hội hiện tại của Tý có là hoàn hảo hay không? Nếu có hãy in ra “YES”, “NO” trong trường hợp ngược lại.

Input Format

Mỗi test bắt đầu bởi 2 số nguyên N và M. M dòng tiếp theo, mỗi dòng gồm 2 số nguyên u, v (u #v) cho biết u và v là kết bạn với nhau trên mạng xã hội của Tý.

Constraints

$1 \leq N, M \leq 100\,000$

Output Format

In ra kết quả của bài toán

Sample Input 0

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

Sample Output 0

```
YES
```

Sample Input 1

```
4 4
```

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

Sample Output 1

NO

[Graph]. Bài 34. Hộp Mặt (SPOJ)

Có K người ($1 \leq K \leq 100$) đứng tại vị trí nào đó trong N địa điểm cho trước ($1 \leq N \leq 1,000$) được đánh số từ $1..N$. Các điểm được nối với nhau bởi M đoạn đường một chiều ($1 \leq M \leq 10,000$) (không có đoạn đường nào nối một điểm với chính nó). Mọi người muốn cùng tụ họp tại một địa điểm nào đó. Tuy nhiên, với các đường đi cho trước, chỉ có một số địa điểm nào đó có thể được chọn là điểm họp mặt. Cho trước K , N , M và vị trí ban đầu của K người cùng với M đường đi một chiều, hãy xác định xem có bao nhiêu điểm có thể được chọn làm điểm họp mặt.

Input Format

Dòng 1: Ghi 3 số: K , N , và M Dòng 2 đến $K+1$: dòng $i+1$ chứa một số nguyên trong khoảng ($1..N$) cho biết địa điểm mà người thứ i đang đứng. Dòng $K+2$ đến $M+K+1$: Mỗi dòng ghi một cặp số A và B mô tả một đoạn đường đi một chiều từ A đến B (cả hai trong khoảng $1..N$ và $A \neq B$).

Constraints

($1 \leq K \leq 100$); ($1 \leq N \leq 1,000$); ($1 \leq M \leq 10,000$);

Output Format

In ra số địa điểm có thể được chọn là điểm họp mặt.

Sample Input 0

```
4 10 15
1457
10 3
5 6
4 9
5 3
8 6
4 6
8 9
3 9
1 9
6 9
7 9
9 6
7 3
2 9
5 9
```


Sample Output 0

2

Sample Input 1

2 4 4

2 3

1 2

1 4

2 3

3 4

Sample Output 1

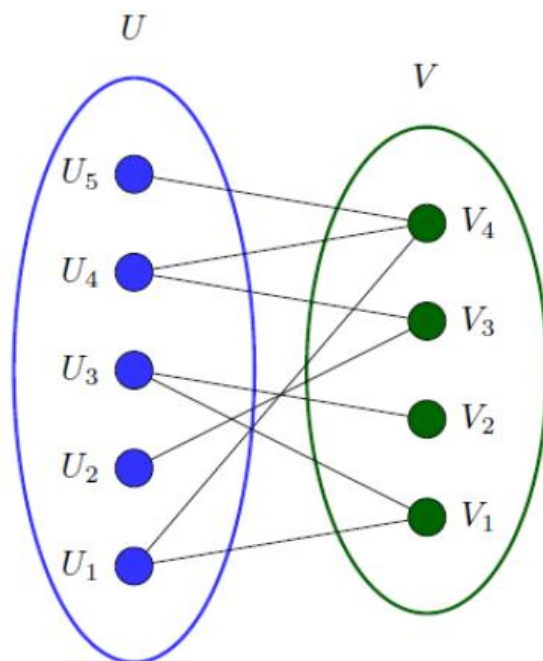
2

Explanation 1

2 điểm có thể hợp mặt : 3, 4

[Graph]. Bài 35. Đồ thị 2 phía (Bipartite graph)

Trong Lý thuyết đồ thị, đồ thị hai phía là một đồ thị đặc biệt, trong đó tập các đỉnh có thể được chia thành hai tập không giao nhau thỏa mãn điều kiện không có cạnh nối hai đỉnh bất kỳ thuộc cùng một tập. Cho đồ thị vô hướng $G = (V, E)$, hãy kiểm tra xem đồ thị đã cho có phải là đồ thị 2 phía hay không?



Input Format

Dòng đầu tiên gồm 2 số n và m tương ứng với số đỉnh và số cạnh của đồ thị. M dòng tiếp theo là cạnh $u-v$, trong đó u khác v .

Constraints

$1 \leq N \leq 1000$; $1 \leq M \leq N \cdot (N-1) / 2$; $1 \leq u, v \leq N$;

Output Format

In ra YES nếu G là đồ thị 2 phía, ngược lại in NO

Sample Input 0

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

Sample Output 0

```
YES
```

Sample Input 1

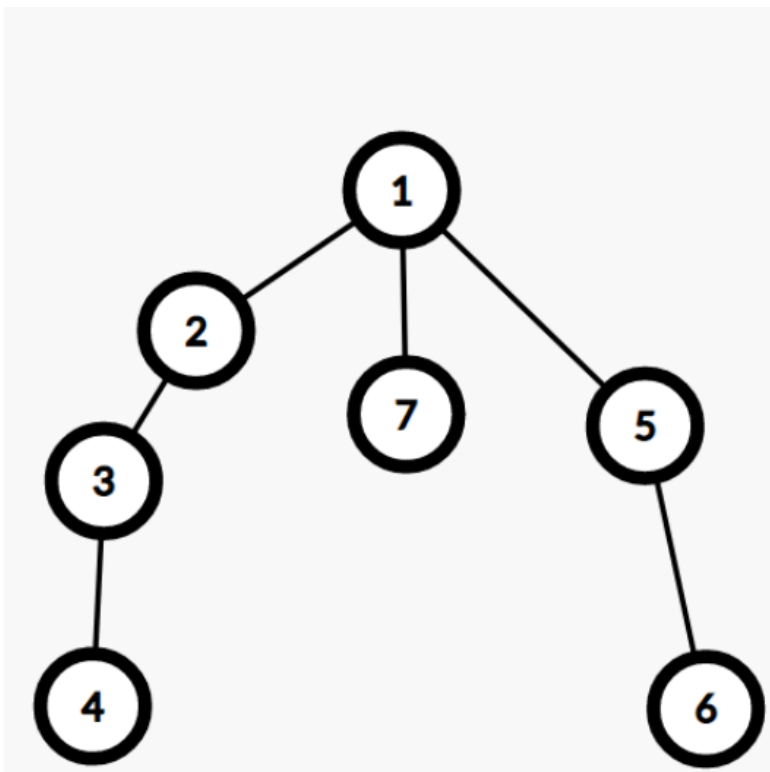
```
6 5
2 1
2 3
3 4
5 4
1 6
```

Sample Output 1

```
YES
```

[Graph]. Bài 36. Độ cao của cây

Cho cây với đỉnh gốc là đỉnh 1, các đỉnh của cây được đánh số từ 1 tới n. Độ cao của một nốt trong cây được tính bằng số cạnh trên đường đi ngắn nhất giữa nốt này tới đỉnh gốc. Như vậy, gốc của cây là đỉnh 1 sẽ có độ cao là 0. Nhiệm vụ của bạn là hãy xác định chiều cao của mọi nốt trong cây. Dưới đây là một ví dụ về cây.

**Input Format**

Dòng đầu tiên là số n - số nốt của cây. N - 1 dòng tiếp theo mỗi dòng mô tả một cạnh của cây.

Constraints $1 \leq n \leq 1000$;**Output Format**

In ra độ cao của các nốt, từ nốt 1 tới nốt n.

Sample Input 0

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

Sample Output 0

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

[Graph]. Bài 37. Quần đảo chìm

Thành phố Vice được xây dựng trên một nhóm các hòn đảo, với những cây cầu nối giữa chúng. Như bất kỳ ai ở Vice City đều biết, nỗi sợ hãi lớn nhất của các phó thành phố là một ngày nào đó các hòn đảo sẽ bị nhấn chìm. Vấn đề lớn của việc này là một khi các hòn đảo bị nhấn chìm, một số hòn đảo khác có thể bị ngắt kết nối. Bạn đã được thị trưởng thành phố Vice thuê để nói cho ông ta biết có bao nhiêu hòn đảo, khi bị nhấn chìm, sẽ ngắt kết nối các bộ phận của thành phố Vice. Bạn nên biết rằng ban đầu tất cả các hòn đảo của thành phố đều được kết nối với nhau.

Input Format

Dòng đầu tiên là N và M, tương ứng với số lượng đảo và cầu; M dòng tiếp theo của mỗi test case mô tả cầu nối giữa 2 hòn đảo u, v;

Constraints $1 \leq N \leq 1000$; $1 \leq M \leq N \cdot (N-1) / 2$; $1 \leq u, v \leq N$;**Output Format**

In ra số lượng hòn đảo khi bị ngập sẽ làm thành phố Vice bị ngắt kết nối.

Sample Input 0

```
6 8
1 3
6 1
6 3
4 1
6 4
5 2
3 2
3 5
```

Sample Output 0

1

[Graph]. Bài 38. Mèo Tom

Chú mèo Tom đang ở gốc 1 cây với nhiều lá và cành. Cái cây này có rất nhiều lá và các đỉnh trung gian. Ở các nốt lá có treo các con cá và mèo Tom mong muốn ăn được nhiều cá nhất có thể vì thế nó dự định sẽ trèo từ gốc cây tương ứng với đỉnh 1 và đi lần lượt tới các nốt lá để ăn cá, nhưng trên đường đi từ nốt gốc lên nốt lá, mèo Tom phải đi qua các nốt trung gian. Các nốt trung gian này sẽ có những con gián và mèo Tom chỉ có thể đi đến các nốt lá để ăn cá nếu trên đường đi từ nốt gốc tới nốt lá có không quá m con gián liên tiếp. Bạn hãy giúp Tom đếm số cá tối đa mà nó có thể ăn được nhé.

Input Format

Dòng đầu tiên là n và m, tương ứng với số lượng nốt của cây và số con gián liên tiếp. Dòng thứ 2 gồm n số có giá trị 1 hoặc 0, tương ứng với vị trí ở nốt thứ i có hoặc không có con gián. N - 1 dòng tiếp theo mỗi dòng là mô tả 1 cạnh của cây.

Constraints

$2 \leq n \leq 1000$; $1 \leq m \leq n$;

Output Format

In ra số con cá tối đa mèo Tom có thể ăn

Sample Input 0

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

Sample Output 0

2

[Graph]. Bài 39. Cây khung DFS

Cho đồ thị vô hướng $G = (V, E)$ liên thông. Hãy in ra các cạnh của cây khung này theo thuật toán DFS bắt đầu từ đỉnh s. Thứ tự mở rộng cây khung của thuật toán DFS luôn mở rộng đỉnh có thứ tự nhỏ hơn trước.

Input Format

Dòng đầu tiên là n, m, và s tương ứng với số đỉnh, số cạnh và đỉnh gốc của cây khung. M dòng tiếp theo mỗi dòng là một cạnh của cây, dữ liệu đảm bảo cạnh chỉ nối giữa 2 đỉnh khác nhau.

Constraints

$1 \leq s \leq n \leq 1000; 1 \leq m \leq n*(n - 1)/2;$

Output Format

In ra các cạnh của cây khung theo thứ tự duyệt bằng thuật toán DFS theo từng dòng.

Sample Input 0

```
7 12 3
6 3
6 2
5 3
5 4
7 3
5 2
7 1
4 3
2 1
6 4
7 2
3 2
```

Sample Output 0

```
3->2
2->1
1->7
2->5
5->4
4->6
```

[Graph]. Bài 40. Cây khung BFS

Cho đồ thị vô hướng $G = (V, E)$ liên thông. Hãy in ra các cạnh của cây khung này theo thuật toán BFS bắt đầu từ đỉnh s . Thứ tự mở rộng cây khung của thuật toán BFS luôn mở rộng đỉnh có thứ tự nhỏ hơn trước.

Input Format

Dòng đầu tiên là n, m , và s tương ứng với số đỉnh, số cạnh và đỉnh gốc của cây khung. M dòng tiếp theo mỗi dòng là một cạnh của cây, dữ liệu đảm bảo cạnh chỉ nối giữa 2 đỉnh khác nhau.

Constraints

$1 \leq s \leq n \leq 1000; 1 \leq m \leq n*(n - 1)/2;$

Output Format

In ra các cạnh của cây khung theo thứ tự duyệt bằng thuật toán BFS theo từng dòng.

Sample Input 0

```
9 18 4
9 3
5 4
```

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

Sample Output 0

```
4->1
4->2
4->3
4->5
4->9
1->6
1->7
5->8
```

[Graph]. Bài 41. Sửa đường

Ở 28techland có n thành phố và m đường giữa chúng. Thật không may, tình trạng của những con đường quá kém nên chúng không thể được sử dụng. Nhiệm vụ của bạn là sửa chữa một số con đường để có một con đường tốt giữa hai thành phố bất kỳ. Đối với mỗi con đường, bạn biết chi phí sửa chữa của nó và bạn nên tìm giải pháp sao cho tổng chi phí càng nhỏ càng tốt.

Input Format

Dòng đầu tiên có hai số nguyên n và m : số thành phố và đường. Các thành phố được đánh số $1, 2, \dots, n$. Sau đó, có m dòng mô tả các con đường. Mỗi dòng có ba số nguyên a , b và c : có một con đường giữa các thành phố a và b , và chi phí sửa chữa của nó là c . Tất cả các con đường đều là đường hai chiều. Mọi con đường đều nằm giữa hai thành phố khác nhau và có nhiều nhất một con đường giữa hai thành phố.

Constraints

$1 \leq n \leq 10^5$; $1 \leq m \leq 2 \cdot 10^5$; $1 \leq a, b \leq n$; $1 \leq c \leq 10^9$

Output Format

In một số nguyên: tổng chi phí sửa chữa tối thiểu. Nếu không có giải pháp nào, hãy in "IMPOSSIBLE".

Sample Input 0

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

Sample Output 0

```
14
```

[Graph]. Bài 42. Lát đường

Ở 28techland có n thành phố và ban đầu không có đường giữa chúng. Tuy nhiên, mỗi ngày một con đường mới sẽ được xây dựng và sẽ có tổng cộng m con đường. Một cụm thành phố là một nhóm các thành phố trong đó có một tuyến đường giữa hai thành phố bất kỳ bằng cách sử dụng các con đường. Sau mỗi ngày, nhiệm vụ của bạn là tìm ra số lượng cụm thành phố và kích thước của cụm thành phố lớn nhất.

Input Format

Dòng đầu tiên có hai số nguyên n và m : số thành phố và đường. Các thành phố được đánh số $1, 2, \dots, n$. Sau đó, có m dòng mô tả các con đường mới. Mỗi dòng có hai số nguyên a và b : một con đường mới được xây dựng giữa các thành phố a và b . Bạn có thể cho rằng mọi con đường sẽ được xây dựng giữa hai thành phố khác nhau.

Constraints

$1 \leq n \leq 10^5$; $1 \leq m \leq 2 \cdot 10^5$; $1 \leq a, b \leq n$;

Output Format

In m dòng: thông tin cần thiết sau mỗi ngày.

Sample Input 0

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

Sample Output 0

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

[Graph]. Bài 43. Tìm đường

Cho một bảng $S[][]$ kích thước $N \times M$, bao gồm các ô trống, các vật cản. Ban đầu bạn ở vị trí S. Nhiệm vụ của bạn là hãy di chuyển tới vị trí T, sao cho số lần đổi hướng không quá hai lần. Các bạn có thể di chuyển từ 1 ô sang 4 ô chung cạnh với ô hiện tại và không có vật cản.

Input Format

Dòng 1 bắt đầu bởi hai số nguyên N và M . N dòng tiếp theo, mỗi dòng gồm M kí tự mô tả bảng S . Trong đó: '.' là một ô trống, '*' là vật cản, 'S' là vị trí xuất phát và 'T' là vị trí đích. (Chỉ có một vị trí S và T duy nhất).

Constraints

$(1 \leq N, M \leq 500)$;

Output Format

In ra "YES" nếu tìm được đường đi, ra in "NO" trong trường hợp ngược lại.

Sample Input 0

```
9 6
**.*..
.**.T
.*..*
.....*
...***
...*S.
*.....
*****.
.....*
```

Sample Output 0

NO

Sample Input 1

```
6 10
S...*...**
...*. *....
.....*.T.
.....*
...*.***.
.....*. *....
```

Sample Output 1

YES

[Graph]. Bài 44. Thuật toán Dijkstra

Cho đồ thị vô hướng có trọng số không âm $G = (V, E)$ được biểu diễn dưới dạng danh sách cạnh trọng số. Hãy viết chương trình tìm đường đi ngắn nhất từ đỉnh s đến tất cả các đỉnh còn lại trên đồ thị. Dữ liệu đảm bảo có đường đi từ đỉnh s tới mọi đỉnh khác trên đồ thị.

Input Format

Dòng đầu tiên là n m và s tương ứng với số lượng đỉnh, cạnh và đỉnh bắt đầu. M dòng tiếp theo là các dòng mô tả cạnh của đồ thị.

Constraints

$1 \leq n \leq 1000$; $1 \leq m \leq n*(n-1)/2$; Trọng số các cạnh là số nguyên không âm không vượt quá 100;

Output Format

In ra đường đi ngắn nhất từ đỉnh u tới mọi đỉnh còn lại

Sample Input 0

```
10 44 5
7 5 60
9 8 31
9 1 83
4 3 25
6 2 1
4 1 52
6 3 76
7 6 95
9 7 84
8 2 91
10 7 8
6 4 32
10 2 76
3 1 62
10 6 88
8 3 12
5 3 15
5 4 40
9 2 20
3 2 5
5 1 36
9 4 98
8 6 65
8 5 95
10 3 55
9 6 95
10 1 5
4 2 70
7 1 80
10 4 35
7 2 99
10 9 24
6 5 94
2 1 77
8 1 12
8 4 76
9 3 27
```

```
5 2 5
9 5 12
10 5 1
8 7 59
6 1 1
10 8 92
7 3 54
```

Sample Output 0

```
6 5 10 35 0 6 9 18 12 1
```

[Graph]. Bài 45. Truy vấn đường đi

Cho đơn đồ thị vô hướng liên thông $G = (V, E)$ gồm N đỉnh và M cạnh, các đỉnh được đánh số từ 1 tới N và các cạnh được đánh số từ 1 tới M . Có Q truy vấn, mỗi truy vấn yêu cầu bạn tìm đường đi ngắn nhất giữa đỉnh X tới Y .

Input Format

- Dòng đầu tiên hai số nguyên N và M ;
- M dòng tiếp theo, mỗi dòng gồm 3 số nguyên u, v, c cho biết có cạnh nối giữa đỉnh u và v có độ dài bằng c .
- Tiếp theo là số lượng truy vấn Q .
- Q dòng tiếp theo, mỗi dòng gồm 2 số nguyên X, Y .

Constraints

$(1 \leq N \leq 100, 1 \leq M \leq N \cdot (N-1)/2); (1 \leq c \leq 1000); Q (1 \leq Q \leq 100000);$

Output Format

Với mỗi truy vấn, in ra đáp án là độ dài đường đi ngắn nhất tìm được

Sample Input 0

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

Sample Output 0

```
8 10 3
```

[Graph]. Bài 46. Di chuyển trên bảng số

Cho một bảng số kích thước $N \times M$. Chi phí khi đi qua ô (i, j) bằng $A[i][j]$. Nhiệm vụ của bạn là hãy tìm một đường đi từ ô $(1, 1)$ tới ô (N, M) sao cho chi phí là nhỏ nhất. Tại mỗi ô, bạn được phép đi sang trái, sang phải, đi lên trên và xuống dưới.

Input Format

- Dòng 1 bắt đầu bởi hai số nguyên N và M .
- N dòng tiếp theo, mỗi dòng gồm M số nguyên $A[i][j]$

Constraints

- $1 \leq N, M \leq 500$.
- $(0 \leq A[i][j] \leq 9)$.

Output Format

In ra một số nguyên là chi phí nhỏ nhất cho đường đi tìm được.

Sample Input 0

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

Sample Output 0

```
28
```

Sample Input 1

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

Sample Output 1

```
24
```

[Graph]. Bài 47. BCDAISY - Chú bò hư hỏng

<https://www.spoj.com/PTIT/problems/BCDAISY/>

Nông dân John có N ($1 \leq N \leq 250$) con bò đánh số từ $1..N$ chơi trên bãi cỏ.

Để tránh bị lạc mất các con bò, mỗi con bò có thể được nối với một số con bò khác bằng dây thừng.

Có tất cả M ($1 \leq M \leq N*(N-1)/2$) dây thừng nối các con bò. Tất nhiên, không có 2 con bò mà có nhiều

hơn 1 dây thừng nối giữa chúng. Dữ liệu cho biết mỗi cặp con bò $c1$ và $c2$ là nối với nhau ($1 \leq c1 \leq N; 1 \leq c2 \leq N; c1 \neq c2$).

Nông dân John buộc cố định con bò 1 bằng sợi dây xích. Các con bò khác phải nối với con bò 1 bằng một số sợi dây thừng. Tuy nhiên, một số con bò hư hỏng không như vậy. Hãy giúp nông dân John tìm các con bò hư hỏng đó (không kết nối tới bò 1). Dĩ nhiên, con bò thứ 1 luôn nối tới chính nó.

INPUT:

* Dòng 1: 2 số nguyên cách nhau bởi dấu cách: N and M

* Lines 2..M+1: Dòng i+1 cho biết 2 con bò nối với nhau bằng sợi dây thứ i là c1 và c2 cách nhau bởi dấu cách.

OUTPUT FORMAT:

* Nếu không có con bò hư hỏng, in ra 0.

* Ngược lại, in ra trên mỗi dòng 1 số nguyên là thứ tự con bò hư hỏng theo thứ tự tăng dần.

SAMPLE INPUT:

```
6 4
1 3
2 3
1 2
4 5
```

SAMPLE OUTPUT:

```
4
5
6
```

Giải thích:

Input có thể mô tả như hình dưới đây:

```
1---2 4---5
 \ |
  \|      6
   \|
    3
```

Để thấy, các con bò 4, 5, và 6 không nối tới con bò 1.

[Graph]. Bài 48. B. Badge

<https://codeforces.com/problemset/problem/1020/B>

In Summer Informatics School, if a student doesn't behave well, teachers make a hole in his badge. And today one of the teachers caught a group of n students doing yet another trick. Let's assume that all these students are numbered from 1 to n. The teacher came to student a and put a hole in his badge. The student, however, claimed that the main culprit is some other student p_a .

After that, the teacher came to student p_a and made a hole in his badge as well. The student in reply said that the main culprit was student p_{p_a} .

This process went on for a while, but, since the number of students was finite, eventually the teacher came to the student, who already had a hole in his badge.

After that, the teacher put a second hole in the student's badge and decided that he is done with this process, and went to the sauna.

You don't know the first student who was caught by the teacher. However, you know all the numbers p_i . Your task is to find out for every student a , who would be the student with two holes in the badge if the first caught student was a .

Input

The first line of the input contains the only integer n ($1 \leq n \leq 1000$) - the number of the naughty students.

The second line contains n integers p_1, \dots, p_n ($1 \leq p_i \leq n$), where p_i indicates the student who was reported to the teacher by student i .

Output

For every student a from 1 to n print which student would receive two holes in the badge, if a was the first student caught by the teacher.

Examples

input

```
3
```

```
2 3 2
```

output

```
2 2 3
```

input

```
3
```

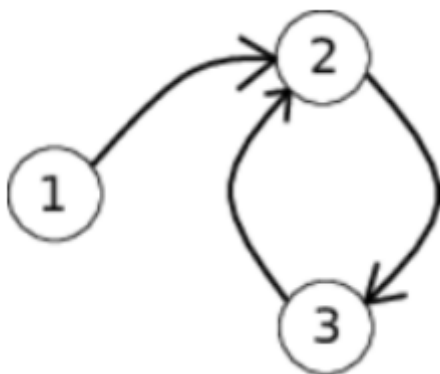
```
1 2 3
```

output

```
1 2 3
```

Note

The picture corresponds to the first example test case.



When $a = 1$, the teacher comes to students 1, 2, 3, 2, in this order, and the student 2 is the one who receives a second hole in his badge.

When $a = 2$, the teacher comes to students 2, 3, 2, and the student 2 gets a second hole in his badge. When $a = 3$, the teacher will visit students 3, 2, 3 with student 3 getting a second hole in his badge.

For the second example test case it's clear that no matter with whom the teacher starts, that student would be the one who gets the second hole in his badge.

[Graph]. Bài 49. PTIT121K - Đường đi lớn nhất

<http://www.spoj.com/PTIT/problems/PTIT121K/>

Sau những tiết học ban đầu hứng thú với môn Điện tử số về hệ cơ số, MĐ dần thấy nản khi phải đối mặt với các loại mạch và cổng @@. Đầu óc cứ nghĩ đến mấy cái hệ cơ số, MĐ lại nghĩ ra một bài toán khác để thử tài các bạn SV PTIT :D. Bài toán như sau:

Cho một bảng vuông $(n \times n)$ ô ($2 \leq n \leq 100$) các ô ghi các số là 0 hoặc 1. Bạn hãy tìm đường đi từ góc trái trên xuống góc phải dưới theo nguyên tắc chỉ được dịch chuyển sang phải và xuống dưới sao cho các số trên đường đi tạo thành một số nhị phân có giá trị lớn nhất.

Input

- Dòng đầu tiên ghi giá trị n
- n dòng tiếp theo, trên mỗi dòng ghi n số 0 hoặc 1 các số này cách nhau ít nhất một khoảng trắng.

Output

- Một số duy nhất là giá trị trong **hệ cơ số 16** của số nhị phân được tạo thành ở trên.

Example

Input:

```
5
1 0 1 1 0
0 0 1 0 1
0 0 1 0 1
1 0 0 1 1
1 1 0 1 0
```

Output:

```
176
```

[Graph]. Bài 50. PTIT018L - ACM PTIT 2018 L - HỆ CƠ SỐ

Cho số nguyên dương N và hệ cơ số B . Hàm $F(B, N)$ được định nghĩa như sau:

Nếu N được biểu diễn trong hệ cơ số B có dạng:

$$N = a_0 + a_1 \cdot b + a_2 \cdot b^2 + \dots$$

thì
$$F(B, N) = a_0^2 + a_1^2 + a_2^2 + \dots$$

Hãy tính giá trị hàm F với mỗi cặp B và N cho trước.

Input

Dòng đầu ghi số lượng bộ test, không quá 1000.

Mỗi bộ test gồm 2 số nguyên: B ($3 \leq B \leq 16$) và số nguyên dương N (trong phạm vi số nguyên không dấu 32 bit).

Output

Với mỗi bộ test, ghi ra trên một dòng giá trị hàm F tính được.

Example

Input:

```
3
10 1234
3 98765
16 987654321
```

Output:

```
30
19
696
```

Dòng đầu tiên là số lượng bộ test T ($T \leq 100$).

Mỗi test gồm dòng thứ nhất chứa số nguyên N là số lượng phần tử của mảng A[] ($1 \leq N \leq 1000$).

Dòng tiếp theo gồm N số nguyên A[i] ($1 \leq A[i] \leq 5000$).

<http://www.spoj.com/PTIT/problems/BCISLAND/>

<http://www.spoj.com/PTIT/problems/BCROBOT/>

<http://www.spoj.com/PTIT/problems/PTIT121G/>

<http://www.spoj.com/PTIT/problems/PTIT122L/>

<http://www.spoj.com/PTIT/problems/PTIT122B/>

<http://www.spoj.com/PTIT/problems/PTIT124E/>

<http://www.spoj.com/PTIT/problems/PTIT125D/>

<http://www.spoj.com/PTIT/problems/PTIT126H/>

<http://www.spoj.com/PTIT/problems/PTIT128A/>

<http://www.spoj.com/PTIT/problems/PTIT128F/>

<http://www.spoj.com/PTIT/problems/STRBALA2/>

<http://www.spoj.com/PTIT/problems/CRAZYFEN/>

<http://www.spoj.com/PTIT/problems/PTIT133H/>

<http://www.spoj.com/PTIT/problems/PTIT134B/>

<http://www.spoj.com/PTIT/problems/PTIT134D/>

<http://www.spoj.com/PTIT/problems/PTIT134I/>

<http://www.spoj.com/PTIT/problems/P145PROB/>

<http://www.spoj.com/PTIT/problems/P146PROA/>

<http://www.spoj.com/PTIT/problems/P147PROF/>

<http://www.spoj.com/PTIT/problems/P148PROD/>
<http://www.spoj.com/PTIT/problems/P153PROD/>
<http://www.spoj.com/PTIT/problems/P167PROB/>
<http://www.spoj.com/PTIT/problems/BCIShAND/>
<http://www.spoj.com/PTIT/problems/BCR&-BOT/>
<http://www.spoj.com/PTIT/problems/PTIT121K/>
<http://www.spoj.com/PTIT/problems/PTIT121G/>
<http://www.spoj.com/PTIT/problems/PTIT122L/>
<http://www.spoj.com/PTIT/problems/PTIT122B/>
<http://www.spoj.com/PTIT/problems/PTIT124E/>
<http://www.spoj.com/PTIT/problems/PTIT125D/>

<http://www.spoj.com/PTIT/problems/PTIT126H/>
<http://www.spoj.com/PTIT/problems/PTIT128A/>
<http://www.spoj.com/PTIT/problems/PTIT128F/>
<http://www.spoj.com/PTIT/problems/STRBALA2/>
<http://www.spoj.com/PTIT/problems/CRAZYFEN/>
<http://www.spoj.com/PTIT/problems/PTIT133H/>
<http://www.spoj.com/PTIT/problems/PTIT134B/>
<http://www.spoj.com/PTIT/problems/PTIT134D/>
<http://www.spoj.com/PTIT/problems/PTIT134I/>
<http://www.spoj.com/PTIT/problems/P145PROB/>
<http://www.spoj.com/PTIT/problems/P146PROA/>
<http://www.spoj.com/PTIT/problems/P147PROF/>
<http://www.spoj.com/PTIT/problems/P148PROD/>
<http://www.spoj.com/PTIT/problems/P153PROD/>
<http://www.spoi.com/PTIT/problems/P167PROB/>