Problem A.

Given a relation $R = \{(a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n)\}$ consisting of n tuples, check if R is reflexive, symmetric and transitive

Input

The first line consists of a single integer n the number of tuples in the relation R.

The next n lines consist of two integers (a, b) representing a tuple in R.

Output

Print 3 strings each being YES/NO, the first one indicating if it is reflexive, second one for symmetric and third one if R is transitive or not.

Example

test	answer
6 1 1 2 2 3 3 1 2 1 3 2 3	YES NO YES
6 1 1 2 2 3 3 1 2 2 1 2 3 3 2	YES YES NO

Note

The first example is reflexive and transitive but not symmetric.

The second example is reflexive and symmetric but not transitive.

Problem B.

Given a relation $R = \{(a_1, b_1),\ (a_2, b_2),\ \ldots,\ (a_n, b_n)\}$ consisting of n tuples, find the transitive closure of R.

Input

The first line consists of a single integer n - the number of tuples in the relation R.

The next n lines consist of two integers (a, b) representing a tuple in R.

Output

First print the number of tuples added in the transitive closure, followed by the tuples added to the closure.

Examples

test	answer
6 1 1 2 2 3 3 1 2 2 1 2 3 3 2	1 1 3
6 1 1 2 2 3 3 1 2 1 3 2 3	0

Problem C.

Let G be a graph containing n vertices labelled from 0 to $n-1$. You are given an array a of length n satisfying the following conditions $\forall i$ from 0 to $n-1$:

- $a_i \in \{0, 1\}$
- If $a_i = 0$, the node labelled i is colored red.
- If $a_i = 1$, the node labelled i is colored blue.

Check if the array a represents a valid 2-coloring of G.

Input

The first line of the input contains 2 integers - n and m, representing the number of nodes and the number of edges of G respectively. The second line contains n integers which represent the elements of array a. Each of the next m lines contains 2 integers which represent an edge of the given graph.

Output

The output should consist of just one line - print **YES** if the array a represents a valid 2-coloring of G, and **NO** otherwise. (Print it in the exact same format, the output will be considered **case-sensitive**).

Examples

test	answer
4 4 0 1 1 0 0 1 0 2 1 3 2 3	YES
4 4 0 1 1 1 0 1 0 2 1 3 2 3	NO

Problem D.

You are given a **simple** graph containing n vertices and m edges. The vertices are numbered from 1 to n. Find the minimum number of edges that need to be added to make this graph connected.

Input

The first line contains 2 integers - the value of n and m respectively. Each of the next m lines contains 2 integers which represent an **undirected** edge of the given graph.

Output

The output should consist of a single line - the minimum number of edges to be added to make the graph connected.

Examples

test	answer
5 3 1 2 1 3 4 5	1
5 4 1 2 1 3 1 4 1 5	0

Problem E.

You are given a **tree** containing n vertices labelled from 1 to n. The tree is rooted at the vertex 1. Find the height of the tree.

Note

The height of a rooted tree is the length of the longest path from the root to any vertex.

Input

The first line contains the value of n. Each of the next $n-1$ lines contains 2 integers which represent an **undirected** edge of the tree.

Output

The output should consist of a single line - print the height of the tree rooted at vertex 1.

Examples

test	answer
4 1 2 1 3 2 4	2
4 1 2 1 3 1 4	1

Problem F.

You are given a tree consisting of n vertices numbered from 1 to n. Find the diameter of the tree. The diameter of the tree is defined as the length of the longest path between any two vertices in the tree.

Input

The input consists of n lines. The first line contains the value of n, the number of vertices in the tree. Each of the next $n-1$ lines contains 2 integers which represent an **undirected** edge of the tree.

Output

Print a single integer - the dimater of the tree.

Examples

test	answer
5 1 2 1 3 3 4 3 5	3
10 6 4 1 3 10 8 9 3 2 7 5 4 2 4 8 5 9 5	6

Note

In the first example the diameter is corresponds to the path $2-1-3-5$

In the second example the diamter corresponds to the path $1-2-9-5-4-2-7$

Problem G.

Given a directed graph consisting of n vertices labeled 1 to n and m edges, report report the vertices of a cycle in G in the cyclic order (if a cycle exist). Otherwise print -1. If there are multiple cycles print the vertices of any of them

Note that the graph is represented using an edge list.

Input

The first line of the input consists of two integers - n and m where n and m denote the number of nodes and edges in the graph.

Then m lines follow, each line consists of two integers u and v denoting an directed edge from u to v.

Output

If there exists a cycle, print the vertices of the cycle in the cyclic order, if there are no cycles print -1. If there are multiple cycles, print any.

Examples

test	answer
5 6 1 2 1 3 2 4 3 4 4 5 5 2	2 4 5
5 5 1 2 1 3 2 4 3 4 4 5	-1

Note

For the first example $4-5-2$, $5-2-4$ are also valid cyclic ordering, however $5-4-2$ is not valid as it is a directed graph and there are no edges from 5 to 4, 4 to 2, 2 to 5 to form a cycle

Problem H.

Given 2 simple graphs G_1 and G_2, check if they are **isomorphic** or not. Note that simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic iff $\exists$ a bijection $f : V_1 \to V_2$ such that $\forall a, b \in V_1$, a and b are adjacent in G_1 iff $f(a)$ and $f(b)$ are adjacent in G_2.

Hint: Check out `next_permutation` function in C++ to generate permutations of an array

Note

Consider the nodes to be labelled with $1-$indexing.

Input

The first line of the input consists of two integers - n_1 and m_1 - the number of nodes and edges respectively in the graph G_1. Then m_1 lines follow, each line consisting of two integers u and v denoting an undirected edge of graph G_1. The next line of the input consists of two integers - n_2 and m_2 - the number of nodes and edges respectively in the graph G_2. Then m_2 lines follow, each line consisting of two integers u and v denoting an undirected edge of graph G_2.

Output

The output should consist of just one line - print **YES** if the two graphs are isomorphic, and **NO** otherwise. (Print it in the exact same format, the output will be considered **case-sensitive**).

Example

test	answer
4 3 1 2 2 4 1 3 4 3 1 4 2 4 2 3	YES
4 3 1 2 2 4 1 3 4 3 2 1 2 4 2 3	NO

Problem I.

You are given a directed graph consisting of n nodes numbered 1 to n and m edges. A node is said to be safe if either there is no outgoing edge or all the outgoing edges incident with a safe node.

More formally a node u is said to be safe if one of the following conditions is satisified -

- There exists no edge (u, v)
- For all edges of the form (u, v), v is a safe node.

Find all the safe nodes in the graph and print them in ascending order.

Input

The first line contains 2 integers - the value of n and m respectively. Each of the next m lines contains 2 integers - u, v. (u, v) represents an directed edge of the graph from u to v.

Output

The output should consist of a two lines - the first line should be the number of safe nodes.

The second line should consist of all the safe nodes in sorted order.

Examples

test	answer
5 6 2 1 3 1 4 2 3 4 4 5 5 3	2 1 2
7 8 1 2 2 3 3 1 3 4 5 4 5 6 6 7 7 8	1 4

Problem J.

Given a **directed** graph G with n vertices and m edges such that any vertex in G has **atmost one** outgoing edge. Find the length of the longest cycle in G.

Note

The length of a cycle is the number of edges present in the cycle.

Input

The first line contains 2 integers - the value of n and m respectively. Each of the next m lines contains 2 integers which represent a **directed** edge of the given graph. It is guaranteed that any vertex has atmost one outgoing edge.

Output

The output contains a single integer - the length of the longest cycle in G.

Examples

test	answer
7 7 1 2 2 3 3 4 4 5 5 2 6 7 7 6	4
2 1 2 2 1	2