Problem A.

Given a degree sequence d_1, d_2, d_3, ... d_n in non-increasing order. Check if it is possible to construct a simple graph that matches this degree sequence.

Hint : Use the *Havel-Hakimi* theorem.

Input

Input will consist of 2 lines. The first line contains n, the length of the degree sequence. The second line contains n integers which represent the degree sequence in non-increasing order.

Output

The output should consist of just one line - print **YES** if it is possible to construct a simple graph that matches this degree sequence, and **NO** otherwise. (Print it in the exact same format, the output will be considered **case-sensitive**).

Example

test	answer
5 4 3 3 3 3	YES
4 4 1 1 1	NO

Problem B.

Given a Hasse diagram consisting of n nodes numbered from 1 to n and two nodes x and y ($1 \leq x, y \leq n$), find the *greatest lower bound* and the *least upper bound* of x and y.

The Hasse diagram is as a directed graph represented using a directed edge list, (u, v) represents a directed edge from u to v

Input

The first line of the input consists of four integers - n, m, x and y where n and m denote the number of nodes and edges in the Hasse diagram whereas x and y denote the nodes of which the greatest lower bound and the least upper bound is to be found.

Then m lines follow, each line consists of two integers u and v denoting the *from* and *to* node of the edge.

Output

Print two integers - the *greatest lower bound* and the *least upper bound* of x and y.

Examples

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

Problem C.

You are given a tree consisting of n vertices numbered from 1 to n. Given a start vertex u and a destination vertex v, print the simple path between these 2 vertices.

A simple path is a path in a graph which does not have repeating vertices.

Input

The input consists of $n+1$ 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. The $n+1^{th}$ line contains 2 integers - the start vertex u and the destination vertex v.

Output

The output contains 2 lines. The first line contains the value of l, the number of nodes in the simple path from u to v. The second line contains a node sequence of length l, representing the simple path from u to v.

Examples

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

Problem D.

Given an undirected graph consisting of n vertices and m edges labeled 1 to n, find the length of the shortest path from vertex 1 to vertex n. The graph is represented by 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 undirected edge between u and v.

Output

Print a single integer - the length of the shortest path.

Example

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

Problem E.

You are given a **simple** graph consisting of n vertices and m edges. The vertices are numbered from 1 to n. Check if the given graph is bipartite or not.

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 just one line - print **YES** if the given graph is bipartite, and **NO** otherwise. (Print it in the exact same format, the output will be considered **case-sensitive**).

Examples

test	answer
5 4 1 4 1 5 4 2 3 5	YES
3 3 1 2 2 3 3 1	NO

Problem F.

BITS offers n courses, some of which have prerequisites. Your task is to determine whether it is possible for a student to take all the courses by scheduling them in a valid order while satisfying the prerequisites.

Input

The first line of the input consists of a single integer - n the number of courses.

Then n lines follow, the i^{th} line consist of an integer k_i the number of pre-requisite courses for the i^{th} course followed by k_i integers - each representing a pre-requisite course.

Output

If it is possible to schedule the courses, print any valid order of the courses. If there is no possible schedule print -1.

Examples

test	answer
6 2 5 6 2 4 5 1 6 2 3 5 0 0	5 6 1 3 4 2
5 1 2 1 3 1 4 1 5 1 1	-1

Problem G.

You are given a **simple** graph containing n vertices and m edges. The vertices are numbered from 1 to n. Count the number of connected components in this graph. *A connected component of a graph G is a connected subgraph of G that is not a proper subgraph of another connected subgraph of G.*

Hint : Use *BFS* or *DFS* to find a connected component.

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 - print the number of connected components in the graph.

Examples

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

Problem H.

Given an undirected graph consisting of n vertices labeled 1 to n and m edges, report if there is a cycle. If there is one, print YES else print NO. The graph is represented using by an edge list.

Hint: Use BFS/DFS to check for a cycle.

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 undirected edge between u and 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.

Example

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

Explanation

For the 1^{st} testcase, $5-8-6-3$, $8-6-3-5$, $6-3-5-8$, $6-8-5-3$, $3-6-8-5$, $5-3-6-8$ and $8-5-3-6$ are also considered valid cyclic sequences.

Problem I.

You are given a **weighted simple** graph with n vertices and m edges. The vertices are numbered from 1 to n. Find the cost of Minimum Spanning Tree of this graph using **Prim's** algorithm. *The cost of a spanning tree is the sum of weights of its edges.*

Input

The first line contains 2 integers - the value of n and m respectively. Each of the next m lines contains 3 integers - u, v and w. (u, v) represents an undirected edge of the graph, and w represents the weight associated with that edge.

Output

The output contains a single integer - the cost of the MST of the graph.

Examples

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

Problem J.

Given a weighted graph G with n vertices labeled 1 to n and m edges. Find the cost of the MST corresponding to this graph using **Kruskal's** algorithm. 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.

Each of the next m lines contains 3 integers - u, v and w. (u, v) represents an undirected edge of the graph, and w represents the weight associated with that edge.

Output

The output contains a single integer - the cost of the MST of the graph.

Examples

test	answer
6 8 1 2 4 1 3 8 2 3 11 2 4 8 3 5 7 3 6 1 4 5 2 5 6 6	21