Problem A. Max flow

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

You are given a network consisting of n nodes and m pipes. The water can flow through these pipes. Each pipe is associated with a non-negative integer capacity denoting the maximum amount of water that may flow through this pipe in a unit of time.

For each node except the source node 1 and the sink node n the Kirchoff's law should be fulfilled: the total amount of all incoming flow should be equal to the total amount of all outcoming flow.

The water flow in each pipe may go in either directions (but the absolute value of the flow should not exceed the capacity of the pipe).

Your task is to find the maximum flow value between the source and the sink and the exact value and the direction of the flow for each pipe.

Input

The first line of the input contains a single integer n ($2 \le n \le 100$), the number of nodes.

The second line of the input contains a single integer m ($1 \le m \le 5000$), the number of pipes.

Each of the following m lines contains three integers a_i, b_i, c_i ($1 \le a_i, b_i \le n$, $0 \le c_i \le 10^4$), the endpoints of i-th pipe and its capacity.

Output

In the first line output the maximum flow between the source and the sink.

In the following m lines output the flow through each of the pipes. The i-th line should contain the value f_i ($-c_i \le f_i \le c_i$), the absolute value of f_i should be equal to the value of the flow through the i-th pipe, the positive sign means flow from a_i to b_i and the negative sign means the flow in the opposite direction.

Examples

standard input	standard output
2 2 1 2 1 2 1 3	4 1 -3

Problem B. Min cut

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

You are given an undirected graph consisting of n vertices and m edges. Find the minimum cut between $s = 1$ and $t = n$.

Input

The first line of the input contains a single integer n ($2 \leq n \leq 100$), the number of vertices.

The second line of the input contains a single integer m ($1 \leq m \leq 400$), the number of edges.

There are no loops nor parallel edges.

Output

In the first line output the number of edges in the minimum cut you have found and the capacity of that cut.

In the following line output the increasing sequence of 1-based indices of the edges forming the cut.

Examples

standard input	standard output
3 3 1 2 3 1 3 5 3 2 7	2 8 1 2

Problem C. Decomposition of flow

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

You are given directed graph. Each edge has integer capacity. You have to find max floe from vertex number 1 to vertex number n and decompose it as union of paths.

Input

The first line contains numbers n ($2 \le n \le 500$) and m ($1 \le m \le 10\,000$) — number of vertices and edges in teh graph ($1 \le n, m \le 25\,000$). i-th of next m lines contains three integers a_i, b_i, c_i means edge from a_i to b_i with capacity c_i ($0 \le c_i \le 10^9$).

Output

Output number of paths. Each of the next lines should contain description of pathes. Each path is described as *value* k $e_1, e_2, \dots e_k$ (value of path, number of edges in the path, indices of the edges of the path). Edges are numbered from 1 to m in order they are given.

In each of the rows you should output edge indices in the order of edges follow in the corresponding path.

Examples

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

Problem D. Cards

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

You have n two-sided cards. On each side of each card there is a single English lowercase letter. You also have a string S consisting of lowercase English letters. You are curious if it is possible to put some cards on a table so that they form a word S (you may choose a side for each of card).

Input

The first line of the input contains a single integer n ($1 \leq n \leq 10^5$), the number of cards.

Each of the following n lines contains two lowercase English letters f_i and b_i, the face side letter of the i-th card and the back side letter of the i-th card.

The last line contains a string S ($1 \leq |S| \leq 10^5$), the desired string.

Output

If it is possible to form a string S by using some of the cards, output $|S|$ integers. The absolute value of the i-th integer should be equal to the 1-based index of a card containing the i-th character of the string. Positive value corresponds to the face side of the card and negative value corresponds to the back side.

If it is impossible to form a string, output "`IMPOSSIBLE`" (without the quotes).

Examples

standard input	standard output
3 ab ab bc bca	-1 -3 2
3 ab ab bc bcc	IMPOSSIBLE

Problem E. Cows

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

You are a farmer and you have an infinite number of cows. You have to get them to the special cow show in the neighbouring city. Unfortunately, the only way you can get there lies through the huge swamp.

You are standing at point (sx, sy) on a stable ground. Your goal is to send as many cows as possible to the destination point (tx, ty) that is also a stable ground (the start point and the end point differ). The only way you can move cows is to use the wooden bridges between the stones in the swamp. Each endpoint of the bridge is either a stone, a starting point or a destination point. Each bridge may be traversed in both directions.

The main difficulty is that a stone disappears in the swamp after a goes through it. Your task is to calculate maximum number of cows that can get from the starting point to the destination point by using the wooden bridges. Cow may enter the bridge only at its endpoint, it cannot jump from bridge to bridge otherwise than by stepping on the stone they share.

Input

The first line of the input contains a single integer n ($1 \le n \le 1000$), the number of bridges.

Each of the following n lines contains four 32-bit signed integers $x1_i, y1_i, x2_i, y2_i$ the coordinates of the endpoints of the i-th bridge. Two endpoints of each bridge are distinct.

There will be no more than one bridge between any pair of points.

The last line contains four 32-bit signed integers sx, sy, tx, ty.

It is guaranteed that there exists no bridge directly connecting the starting point and the destination point.

Output

Output the maximum number of cows that you can send to the destination point.

Examples

standard input	standard output
8 0 0 1 0 1 0 2 1 1 0 2 -1 2 1 3 0 2 -1 3 0 1 0 4 0 3 0 4 0 0 0 3 0 0 0 4 0	2

Problem F. Matan

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

In university of city M an experiment takes place. Tutors decide what to study during the course of lections.

One tutor studying calculus associated each theme of the course with an integer usefulness (possibly negative). Tutor wants to maximize total usefulness of all themes, though it's not easy. To make students understand anything you need to tell them some other themes since proofs of facts may use another facts as a pre-requisite.

For each theme you know the list of themes that it depends from. The cycles on dependencies are allowed, but in order to fully understand the theme you have to tell students all themes from the cycle (in the other words, the order of telling is not significant but the overall set of themes should satisfy all dependencies).

You have to find the set of themes that maximizes the total usefulness.

Input

The first line contains n ($1 \le n \le 200$), the number of themes. The second line contains n integers between -1000 and 1000, the usefulnesses of all themes.

The following n line describe dependencies between themes. The i-th line starts with k_i, the number of themes students should know in order to understand the i-th theme. After that k_i distinct integers between 1 and n follow, the indices of dependency themes.

The total number of all dependencies doesn't exceed 1800.

Output

Output the only number: the maximum total usefulness of all chosen themes.

Examples

standard input	standard output
4 -1 1 -2 2 0 1 1 2 4 2 1 1	2
3 2 -1 -2 2 2 3 0 0	0

Problem G. Perspective

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Breaking news! A Russian billionaire has bought a yet undisclosed NBA team. He's planning to invest huge effort and money into making that team the best. And in fact he's been very specific about the expected result: the first place.

Being his advisor, you need to determine whether it's possible for your team to finish first in its division or not.

More formally, the NBA regular season is organized as follows: all teams play some games, in each game one team wins and one team loses. Teams are grouped into divisions, some games are between the teams in the same division, and some are between the teams in different divisions.

Given the current score and the total number of remaining games for each team of your division, and the number of remaining games between each pair of teams in your division, determine if it's possible for your team to score at least as much wins as any other team in your division.

Input

The first line of the input file contains N ($2 \le N \le 20$) — the number of teams in your division. They are numbered from 1 to N, your team has number 1.

The second line of the input file contains N integers w_1, w_2, ..., w_N, where w_i is the total number of games that i^{th} team has won to the moment.

The third line of the input file contains N integers r_1, r_2, ..., r_N, where r_i is the total number of remaining games for the i^{th} team (including the games inside the division).

The next N lines contain N integers each. The j^{th} integer in the i^{th} line of those contains a_{ij} — the number of games remaining between teams i and j. It is always true that $a_{ij} = a_{ji}$ and $a_{ii} = 0$, for all i $\sum_j a_{ij} \le r_i$.

All the numbers in the input file are non-negative and don't exceed 10 000.

Output

On the only line of output, print "`YES`" (without quotes) if it's possible for the team 1 to score at least as much wins as any other team of its division, and "`NO`" (without quotes) otherwise.

Example

standard input	standard output
3 1 2 2 1 1 1 0 0 0 0 0 0 0 0 0	YES
3 1 2 2 1 1 1 0 0 0 0 0 1 0 1 0	NO

Problem H. Broken Parquet

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

The floor of the rectangular room of size $n \times m$ is filled with the parquet. Some of the cells of the room floor are broken and Petr decided to fix them.

There are tiles of two types in the store. Tile of the first type is a rectangle 1×2 (it can be rotated and used as a rectangle 2×1) and it costs a coins. Tile of the second type is a rectangle 1×1 and it costs b coins. The tile of the first type cannot be split into pieces 1×1. Tiles should cover all the broken cells, also they should not overlap.

Find out the minimum number of coins Petr has to spend in order to fix the floor. Everything happens in a strange parallel universe, so a and b may be negative.

Input

The first line of the input contains 4 integers n, m, a and b ($1 \le n, m \le 300$, $-1000 \le a, b \le 1000$). Each of the following n lines contains m characters: "." means the whole cell of the floor and "*" means the broken cell that should be covered with a tile.

Output

Output the minimum number of coins Petr has to spend in order to fix the floor.

Example

standard input	standard output
2 3 3 2 .** .*.	5

Problem I. Full orientation

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	256 mebibytes

You are given undirected graph without loops and multiedges. You have to make this graph directed in such a way, that maximal outgoing degree is minimal possible.

Input

The first line contains numbers n and m — number of vertices and edges in teh graph ($1 \le n, m \le 25\,000$). Next m lines contain pairs of integers from 1 to n — edges of the graph.

Output

Output m integers from 0 to 1. If i-th edge is given as a_i, b_i then zero means it should go from a to b, and one means edge from b to a.

Examples

standard input	standard output
4 4 1 2 1 3 4 2 4 3	1 0 1 1 0
5 5 1 2 2 3 3 1 1 4 1 5	1 0 0 0 1 1

Problem J. Binary Tree on Plane

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	256 mebibytes

A root tree is a directed acyclic graph that contains one node (root), from which there is exactly one path to any other node.

A root tree is binary if each node has at most two outgoing arcs.

When a binary tree is painted on the plane, all arcs should be directed from top to bottom. That is, each arc going from u to v must meet the condition $y_u > y_v$.

You've been given the coordinates of all tree nodes. Your task is to connect these nodes by arcs so as to get the binary root tree and make the total length of the arcs minimum. All arcs of the built tree must be directed from top to bottom.

Input

The first line contains a single integer n ($2 \le n \le 400$) — the number of nodes in the tree. Then follow n lines, two integers per line: x_i, y_i ($\|x_i\|, \|y_i\| \le 10^3$) — coordinates of the nodes. It is guaranteed that all points are distinct.

Output

If it is impossible to build a binary root tree on the given points, print "-1". Otherwise, print a single real number — the total length of the arcs in the minimum binary tree. The answer will be considered correct if the absolute or relative error doesn't exceed 10^{-6}.

Examples

standard input	standard output
3 0 0 1 0 2 1	3.650281539872885
4 0 0 1 0 2 1 2 0	-1

Problem K. Hard Life

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

John is a Chief Executive Officer at a privately owned medium size company. The owner of the company has decided to make his son Scott a manager in the company. John fears that the owner will ultimately give CEO position to Scott if he does well on his new manager position, so he decided to make Scott's life as hard as possible by carefully selecting the team he is going to manage in the company. John knows which pairs of his people work poorly in the same team. John introduced a hardness factor of a team — it is a number of pairs of people from this team who work poorly in the same team divided by the total number of people in the team. The larger is the hardness factor, the harder is this team to manage. John wants to find a group of people in the company that are harderst to manage and make it Scott's team. Please, help him.

Input

The first line of the input file contains two integer numbers n and m ($1 \le n \le 100$, $0 \le m \le 1000$). Here n is a total number of people in the company (people are numbered from 1 to n), and m is the number of pairs of people who work poorly in the same team. Next m lines describe those pairs with two integer numbers a_i and b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$) on a line. The order of people in a pair is arbitrary and no pair is listed twice.

Output

Write to the output file an integer number k ($1 \le k \le n$) — the number of people in the hardest team, followed by k lines listing people from this team in ascending order. If there are multiple teams with the same hardness factor then write any one.

Examples

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