Problem A. Just a flow

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

You are given a system of nodes and pipes. Water may flow throw pipes. For each pipe you know maximal speed of the water flow. Each moment of time for each node except source and sink amount of water flowing in the node is equal to amount of water flowing out the node

You have to find maximal amount of water per unit time which can flow between source and sink. Also you have to find speed of the flow along each pipe, which should be to acquire maximal amount.

All pipes can be used in both directions. Between pair of nodes can be more than one pipe.

Input

The first line contain integer N — number of nodes ($2 \le N \le 100$). All nodes are numbered from 1 to N. The source has number one, sink has number N. The first line contain integer M — number of pipes ($1 \le M \le 5\,000$). Next M lines describe pipes. Each pipe is described by three integers A_i, B_i, C_i, where A_i, B_i — indices of nodes, which the pipe connects, and C_i ($0 \le C_i \le 10^4$) — maximal speed of the water flow throw the pipe.

Output

In the first line output maximal amount of water per unit of time. i-th of the next M lines should contain speed throw the pipe number i. If direction of the flow throw the pipe differs from order of nodes in input, output the speed with sign minus; Output numbers with accuracy at least 3 digits after decimal point.

Example

standard input	standard output
2 2 1 2 1 2 1 3	4.0000000 1.0000000 -3.0000000

Problem B. Cut

Input file:	*standard input*
Output file:	*standard output*
Time limit:	0.5 seconds
Memory limit:	256 mebibytes

You are given undirected graph. Find minimal cut between vertices 1 and n.

Input

The first line contain integers n ($1 \le n \le 100$) — number of vertices in the graph, and m ($0 \le m \le 400$) — number of edges in the graph. Next m lines describe the edges. Each edge is described by numbers of its ends and its capacity (positive integer no more than 10 000 000). There are no multiedges and loops.

Output

On the first line output number of edges going throw minimal cut and its summary capacity. Next line should contain increasing sequence of indecies of edges (edges are numbered by integers from 1 to m in order they are given).

Example

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:	2 seconds
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 (equals to the value of max flow). Each of the next lines should contain description of pathes. Each path is described as $value$ k $e_1, e_2, \ldots e_k$ (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.

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. Snails

Input file:	*standard input*
Output file:	*standard output*
Time limit:	0.5 seconds
Memory limit:	256 mebibytes

Two snails Masha and Petya are on the lawn with apricots and want to go back to their home. Lawns are numbered by integers from 1 to n and are connected by roads. Two lawns may be connected by many roads. It may happen, a road connects a lawn with itself. Due to reasons of hygiene, if one snail already went along the road, another one can not use this road. Help to Petya and Masha to get back home.

Input

The first line contain four integer numbers — n, m, s и t (number of lawns, number of roads, index of lawn with apricots, index of lawn with the home). Next m lines describe roads. Each line contain pair of integers (x, y), means there is road from lawn number x to lawn number y. (due to the nature of snails and the space all roads are one-way).

Limitations: $2 \le n \le 10^5, 0 \le m \le 10^5, s \ne t$.

Output

If there is a solution output `YES` and two lines with sequences of indices of lawns in the paths. At first output the path for Masha, then output the path for Petya. If there is no solution output `NO`. If there are several solutions output any of them.

Example

standard input	standard output
3 3 1 3 1 2 1 3 2 3	YES 1 3 1 2 3

Note

You are given oriented graph, find two disjoint by edges paths from s to t. Output vertices of desired paths.

Problem E. Matan

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

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

One tutor specialized on mathematical analysis (Matan) valued all known to him themes in the course. As result each theme has a number (possible negative) — "— usefulness of the topic. Tutor wants to maximize summary usefulness of the themes, but it's not so easy. To make students understand theme, you need to tell them some other themes, beacause of some proofs are based on facts from other themes. But if there is cycle of dependencies, you may read all the themes of the cycle and finally all stdudents will understand all the themes from the cycle.

You have to find set of themes. This set should be clear for students and usefulness of the set should be maximized.

Input

The first line contains integer N ($1 \le N \le 200$) – number of themes. The second line contains N integers, not exceeding 1 000 by absolute value — usefulness of themes. Next N lines describe dependencies between themes. i-th line start with k_i — number of themes students should already know to understand i-th theme. Then k_i diffetent integers from 1 to N – indecies of the themes;

Summary number of dependencies do not exceed 1 800.

Output

Output the only number — maximal summary usefulness of choosen 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 F. 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 $w_1, w_2, \ldots, w_N$, 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 $r_1, r_2, \ldots, r_N$, 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 G. 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 H. Looking for Brides

Input file:	*standard input*
Output file:	*standard output*
Time limit:	0.3.seconds
Memory limit:	256 mebibytes

Once upon a time the king of Flatland decided to send his k sons in a journey, so each of them could find a bride. Everyone knows that Flatland consists of n cities, and some of them are connected by roads. The king lives in the capital, the city number 1, and the city number n is famous for its brides.

Long story short, king's will was that each his son must travel from city 1 to city n by roads. Many brides live in city n, though much less are beautiful, so young princes beware each other. If any son passes through some road then others avoid taking it even afterwards, because there may be traps on it.

The king loves his children and respects their time. He wants to select their paths to minimize the average travel time.

Input

The first line of the input contains three integers n, m, k — the number of cities and roads in Flatland and the number of king's sons ($2 \le n \le 200$, $1 \le m \le 2000$, $1 \le k \le 100$). Next m lines describe roads. Each of them contains three integers — the numbers of cities it connect and its length (not exceeding 10^6). Each road is bidirectional. There may be several roads between any pair of cities.

Output

If it is impossible to send all sons using non-intersecting paths, print -1.

Otherwise print the minimum possible average travel time with absolute or relative precision 10^{-5}. In the next k lines print sons' paths. Each path must follow the format c_i e_1 $e_2 \ldots$, where c_i is the number of roads in the path and e_j are the ids of the roads. Roads are numbered from 1 in the order of input.

Example

standard input	standard output
5 8 2 1 2 1 1 3 1 1 4 3 2 5 5 2 3 1 3 5 1 3 4 1 5 4 1	3.00000 2 3 8 2 2 6