Problem A. The flow in network

Input file: `system input`
Output file: `system output`
Time limit: 2 seconds
Memory limit:

In many problems on graph theory, the concepts of *network* and *flow* are widely used,

Network is a directed graph $G = (V, E)$, where for each edge $(u, v) \in V$ the value $c(u, v) \geq 0$ is given, named *capacity* of the edge. In the case $(u, v) \notin V$ it is convenient to think that $c(u, v) = 0$. In the network there are two special vertices: *source* s and *sink* t.

The *flow* in network G is the function $f : V \times V \to \mathcal{R}$, for which three propery hold:

- *Capacity constraint* — $\forall u, v \in V \ f(u, v) \leq c(u, v)$;
- *Antisymmetry* — $\forall u, v \in V \ f(u, v) = -f(v, u)$;
- *Conservation of flows* — $\forall u \in V \setminus \{s, t\} \ \sum_{v \in V} f(u, v) = 0$.

You are given a network and some function on the pairs of vertices. Check that this function is the flow in the given network.

Input

The first line contains the number of vertices in network N ($2 \leq N \leq 100$). The vertices in the network have ids from 1 to N.

The next N lines contain the capacity of edges. Each such line contains N numbers. The j-th number in $i + 1$-th line defines $c(i, j)$. All capacities are non-negative and are not bigger than 10^4. It is guaranteed that $c(i, i) = 0$.

A single blank line follows.

The next N lines contain the values of the function f in the same format. These values don't exceed 10^4 by the absolute value.

The source of the network is the vertex 1, and the sink is the vertex N.

Output

The first line of the output shoud contain `YES`, if the function is a flow, and `NO`, otherwise.

Examples

system input	system output
4 0 1 3 0 0 0 0 2 0 0 0 4 0 0 0 0 0 1 2 0 -1 0 0 1 -2 0 0 2 0 -1 -2 0	YES
4 0 1 3 0 0 0 0 2 0 0 0 4 0 0 0 0 0 2 1 0 -2 0 0 2 -1 0 0 1 0 -2 -1 0	NO

Problem B. Maximum flow

Input file: `system input`
Output file: `system output`
Time limit: 2 second
Memory limit:

A directed graph is given where each edge has an integer capacity. Find the maximum flow from the vertex 1 to the vertex n.

Input

The first line of the input file contains integers n and m –= the number of vertices and the number of edges, correspondingly ($2 \leq n \leq 100$, $1 \leq m \leq 1000$). The next m lines contain three non-negative integers each describing an edge — the source, the target and the capacity of the edge. The vertices are numbered starting from 1, and the capacities do not exceed 10^5.

Output

Output a single number — the maximum flow from the vertex 1 to the vertex n.

Examples

system input	system output
4 5 1 2 1 1 3 2 3 2 1 2 4 2 3 4 1	3

Problem C. Experimental treatment

Input file: `system input`
Output file: `system output`
Time limit: 2 seconds
Memory limit:

After a lot of unsuccessful attempts to diagnose a desease of a new patient, MD House decided to try a new experimental treatment. During the treatment each hour Forman offered the patient to choose and drink one of two pills. It is known, that right after the n-th pill was chosen, the patient suddenly recovers. The patient remembers how many pills of each type he drinks, and Forman knows all the pairs of pills, which he offered throughout the time. Because you need to know, which pills helped, House wants to reestablish the type of each pill, chosen by the patient. Please, help him.

Input

The first line contains the number of pills, chosen by the patient, n and the number of different types of pills, which are in the hospital, m ($1 \le n \le 1000, 2 \le m \le 1000$). Each i-th line, starting from the second till $(n+1)$-th, contains the pair of integers a_i, b_i ($1 \le a_i, b_i \le m$, $a_i \ne b_i$) — the ids of the types of pills, which were offered by Forman at $(i-1)$-th hour. The last line contains m numbers c_j — the number of chosen pills of the type j ($0 \le c_j \le n$). The types are numerated from 1.

Output

Output the sequence of n numbers, where i-th number is equal to the type of the pill, chosen at i-th hour. If there are multiple answers, output any. If there are no answers, print -1.

Example

system input	system output
3 3 1 2 1 3 2 3 1 2 0	2 1 2
3 3 1 2 1 3 2 3 1 1 0	-1

Problem D. Tea

Input file: `system input`
Output file: `system output`
Time limit: 2 seconds
Memory limit:

In numerous departments of some huge company there are n people working. They enjoy drinking tea during the break. They are disciplined therefore they have only one break daily. To make this break as pleasant as possible each of the employees drinks a tea with one of their favorite flavours. An employee could drink tea with different flavours at different days. The flavours of tea are numbered from 1 to m.

Recently, the employees of the department bought a large set of teabags. This set contains a_1 teabags of the first flavour, a_2 teabags of the second flavour, $\ldots$, and a_m teabags of the m-th flavour. Now they want to know, how long will they have enough teabags, so that each of the employees could drink some tea each day?

Each employee in the department drinks one cup of tea that brews one teabag. Teabags are not brewed again.

Input

The first line of the input file contains two integers n, m ($1 \le n, m \le 50$). The second line contains m integers $a_1, \ldots, a_m$ ($1 \le a_i \le 10^6$ for all i from 1 to m). Each of the following n lines contains description of favorite tastes of employees. The i-th line describes i-th employee in the following format: positive integer k_i — number of favorite varietes following k_i integers from 1 to m — numbers of this varietes.

Output

The sole line of the output should contain the maximum number of days during which employees will have enough sufficient number of bags.

Example

system input	system output
3 3 2 7 4 2 1 2 1 2 2 2 3	4

Problem E. Molecule

Input file:	`system input`
Output file:	`system output`
Time limit:	1 second
Memory limit:	

Arthur and Leonard play in the following game. In some cells of rectangle Arthur draws one of the chemical elements '`H`', '`O`', '`N`' and '`C`', after that Leonard has to draw lines between symbols in neighbouring cells, such that he will get correct molecules. Unfortunately, Arthur likes to draw a huge amount of symbols, while Leonard cannot even understand whether it is possible to draw the lines in the described fashion. Help him to answer on that question.

In this problem the lines between chemical elements create a correct molecule, if they satisfy the following conditions:

- each line connect elements, written in cells, which are neighbours by side;
- between each pair of elements there is at most one line;
- each element should have exactly the defined number of adjacent lines (1 for H, 2 for O, 3 for N, 4 for C);
- empty cells don't have adjacent lines; and
- at least one cell contains an element inside.

Input

The first line contains two integer numbers n and m ($1 \le n, m \le 50$) — the sizes of the rectangle. Then n lines follows, with exactly m symbols in each, defining the distribution of elements in the rectangle; empty cells are denoted by symbol "`.`".

Output

The sole line of the output should contain one word "`Valid`", if it possible to draw lines with described conditions, and "`Invalid`" otherwise.

Example

system input	system output
3 4 HOH. NCOH OO..	Valid
3 4 HOH. NCOH OONH	Invalid

Problem F. Evacuation

Input file:	`system input`
Output file:	`system output`
Time limit:	2 seconds
Memory limit:	

In T minutes the army of Loky will attack Earth. Avengers do not have time to prevent portals in New York from opening, so Captain America decided to evacuate all citizens. He only need to check, if everybody could be in safe position before the start of the invasion.

The surroundings of New York could be represented as a number of small cities, connected with one-directional roads. Each road is characterized by its length and capacity. The length of the road l means, that if you enter it at time t, the car will be at the end after l minutes, in $t + l$. The capacity of road s means, that each minute, at most s cars could enter it. When you get to the city, each car could enter some other road, or it could stop there for any time and then move again.

Captain America have already decided, in which city the citizens have to be. Also, he knows, how many cars there are in the city. Now, he needs to check, if all citizens could evacuate before the invasion. If possible you should find the minimal time which evacuation could take. Otherwise, you should find the minimal number of cars, which will be late, otherwise.

Input

The first line contains four integers n, m, K and T ($1 \le n \times T \le 10\,000$, $1 \le m, K \le 10\,000$) — the number of cities around New York, the number of roads, the number of cars and the time before the invasion, respectively. Next m lines contain the description of roads between cities.

Each road is described with four integers u, v, l and s ($1 \le u, v \le n$, $u \ne v$, $1 \le s \le 3\,000$, $1 \le l \le 200$) — in-point, out-point, its length and capacity, respectively.

There could be only at most one road between two cities. New York has index 1, and the safe city has index n. In time 0 all cars are in New York.

Output

If all citizen could get to the safe city in no longer than T minutes, output the minimal number of minutes, to achieve that. Otherwise, output the minimal number of cars which will be late.

Example

system input	system output
5 5 10 10 1 2 2 2 2 3 1 1 2 4 1 1 4 5 2 4 3 5 2 4	9

Problem G. Non-redundant satisfiability

Input file: `system input`
Output file: `system output`
Time limit: 2 seconds
Memory limit:

A CNF boolean formula is a boolean formula:

$$\Psi(x_1, \ldots x_n) = (t_{11} \vee t_{12} \vee \ldots \vee t_{1s_1}) \wedge \ldots \wedge (t_{k1} \vee t_{k2} \vee \ldots \vee t_{ks_k}),$$

where each term t_{ij} is either a variable x_l or the negation of a variable $\overline{x_l}$. If none of the terms does not appear in a CNF boolean formula then this formula is without repeats. For example, the formula $(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_3 \vee x_4)$ is without repeats, while the formula $(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_3} \vee x_4)$ has a repeat because the term $\overline{x_3}$ appears twice.

The problem about the non-redundant satisfiablity is stated as follows: is it possible to set the values to the variables in such a way that each "bracket" contains exactly one "true" term. For example, the formula $(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_3 \vee x_4)$ has non-redundant satisfiability by $x_1 = 1$, $x_2 = 0$, $x_3 = 1$ and $x_4 = 0$.

You are given a CNF boolean formula without repeats. You have to find whether the formula has the non-redundant satisfiability.

Input

The first line of the input contains two integers n and k ($1 \le n \le 300, 1 \le k \le 300$) — the number of variables and the number of "brackets". Then each of the next k lines contains a description of a bracket: the number of terms s_i followed by the list of terms. Each term is represented by one integer q: if $q > 0$ then the term is x_q, otherwise, the term is $\overline{x_{-q}}$.

Output

If the formula has the non-redundant satisfiability, then the first line should contain "`YES`". The second line should contain n integers, 0 or 1 — the values of $x_1, \ldots, x_n$.

If the formula does not have the non-redundant satisfiability, the sole line of the output should contain "`NO`".

Examples

system input	system output
4 2 3 1 2 -3 3 -1 3 4	YES 1 0 1 0
5 4 3 1 2 3 3 -1 4 5 2 -2 -4 2 -3 -5	NO