Problem A. Maximum Flow

Time limit: 2 seconds
Memory limit: 256 megabytes

You are given a directed graph G. Each edge has some capacity. Find the maximum flow from vertex 1 to vertex n.

Input

The first line of input contains n and m — the number of vertices and edges in the graph ($2 \le n \le 500$, $1 \le m \le 10\,000$). Thw following m lines contain lines. Each edge is described with three integers: the beginning of the edge, the end of the edge and its capacity. Capacities do not exceed 10^9.

Output

First output the maximum flow from 1 to n.

Then for each edge output the flow on the edge.

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 2 1
4 5 1 2 2 1 3 2 2 3 1 2 4 2 3 4 2	4 2 2 0 2 2
4 3 1 2 1 1 3 1 1 4 4	4 0 0 4

Problem B. Minimum Cut

Time limit: 2 seconds
Memory limit: 256 megabytes

You are given an undirected graph. Find the minimum cut between vertices 1 and n.

Input

The first line of input contains an integer n ($2 \le n \le 100$) — the number of vertices of the graph, and m ($0 \le m \le 400$) — the number of edges of the graph.

The following m lines contain edge descriptions. Each edge is described by indices of vertices it connects and its capacity (positive integer not exceeding 10 000 000), no two vertices are connected by more than one edge.

Output

The first line of output must contain the number of edges in the minimum cut and their total capacity. The second line must contain the edges of the cut, listed in increasing order. Edges are indexed in order they are given in the input file.

Example

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

Problem C. Cards

Time limit: 1 second
Memory limit: 256 megabytes

You have n cards, each card has two sides. Each side of a card contains an English letter.

Your friend has given you a string. You would like to create this string by arranging the cards in a row in front of you. You can flip the cards and rearrange them any time you like, in the resulting string each card can be used at most once with any of its sides up.

Solve the problem you set to yourself!

Input

The first line of input contains a single integer n ($1 \le n \le 100\,000$) — the number of cards. The following n lines contain the cards themselves. Each of these lines contain two lowercase English letters, the letter on the first side of the card, and the letter on the second side of the card, respectively.

The following and the last line contains the string given to you by a friend, it is non-empty and consists of at most 100 000 lowercase English letters. Denote the length of the given string as m.

Output

If it is possible to arrange the cards to create the given string, output m integers. The i-th of these integers must be equal to the index of the card that must be placed at the i-th position. If the card must be put with its first side letter up, output just its index as the positive integer. If the card must be flipped to show its second side letter, output its index multiplied by -1.

Otherwise print "`IMPOSSIBLE`".

Examples

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

Problem D. Snails

Time limit: 2.5 seconds
Memory limit: 256 megabytes

Two snails Masha and Petya are located in the graph of size n and want to get home. Vertices of the graph are numbered from 1 to n. There are m unidirectional arcs, there may be multiple arcs between a pair of vertices and there may be an arc that connects vertex with itself. Masha and Petya dislike each other so they would never use the same arc. Help them get home.

Input

The first line of the input contains four integers n, m, a and h ($2 \le n \le 100\,000$, $0 \le m \le 100\,000$, $1 \le a, h \le n$, $a \ne h$) — the number of vertices and edges in the graph, the index of vertex snails are currently located at and the index of the vertex where the home is located.

Then follow m pairs of integers. Pair (x_i, y_i) means that there the i-th arc goes from vertex x_i to vertex y_i.

Output

If there exists a solution print "`YES`" on the first line of the output. Then print two lines containing the paths of the snails (Masha goes first as she is a lady). If there is no solution print "`NO`" on the only line of the output. If there are multiple solution, print any of them.

Example

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

Problem E. Painting

Time limit: 2 seconds
Memory limit: 256 megabytes

You are given a rectangular painting consisting of black and white cells. To make the painting look smooth you have decided to paint gray any border that separates white and black cell.

Before the start of this task you found out that gray paint is very expensive. To save some money you can change the color of some cells (paint some white cells black, and some black cells white) and only then paint grey borders. Of course, black and white paint also costs some money.

Given the description of the initial painting and the cost of white, black and gray paint find out the minimum cost required to finish the painting.

Input

The first line of the input contains five integers n, m, w, b, g ($1 \le n, m \le 70$, $1 \le w, b, g \le 1000$) — height and width of the painting, cost of painting one cell while, painting one cell black and the cost of one gray border line, respectively. Then follow n lines each containing m characters. Character '`B`' stands for the black cell, while character '`W`' stands for the white.

Output

Print one integer — the minimum cost required to finish the painting.

Example

standard input	standard output
3 2 10 12 1 BW WB BW	7

Problem F. Inidividual Trajectory

Time limit:	1 second
Memory limit:	256 megabytes

Flatland University has individual trajectories for its students. There are n topics students can study. For each topic they know its *usefulness* — an integer, probably negative.

The student wants to maximize the total usefulness of topics they would study, but there are some dependencies between topics. For each topic there can be several topics that are required to understand it.

The study plan is the set of topics that the student would study. The study plan is *consistent* if for each topic in the plan all of its required topics are also included into the plan.

Help the student to create consistent study plan with maximum possible total usefulness of included topics.

Input

The first line of input contains one integer — n ($1 \le n \le 200$). The second line contains n integers not exceeding 1 000 — the usefulness of each topic.

The following n lines describe dependencies between topics. Each lines starts with an integer k_i — the number of topics required for the given topic, followed by indices of the required topics.

The sum of k_i doesn't exceed 1 800.

Output

Output one integer — the maximum possible total usefulness of a consistent study plan.

Example

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

Problem G. Trading Game

Time limit:	2 seconds
Memory limit:	256 megabytes

You are playing a trading game over internet. There are n types of resources in the game and m possible buildings that can be bought for resources.

The players make moves in turn. During your move you can trade with other players. When trading you can exchange some resource to other one if the other player agrees. Trade is always one for one.

It's your turn now and you want to build some building. However, you don't have enough resources to build it. You would like to trade with other players. However, each player also wants to build some building. For each player i there is a building v_i she wants to build. The player agrees to make exchange only if the number of resources needed for her building that she is missing decreases after the trade. For example, let there be five types of resources: sheep, brick, wood, ore and wheat. You would like to build a settlement that costs one sheep, one brick, one wood and one wheat. However you only have two sheep and two wood. Let one of your opponents have three bricks and two wheat, and let her want to build road which requires one brick and one wood. Then you can exchange a wood for a wheat or a wood for a brick with her, because now she is missing one resource (wood) for her building, but after exchange she would be missing zero resources. However, she wouldn't agree to exchange sheep for anything because she doesn't need it. You cannot perform both trades above because after the first exchange the player no longer needs wood.

Given a set of resources you and other players have, and the building everyone would like to build, find out whether it is possible to organize trade so that you could build the building you want.

Input

The first line of the input file contains p — the number of players, n — the number of resources, m — the number of buildings ($1 \le p, n, m \le 20$). The second line contains n words — the names of resources.

The following m lines contain descriptions of buildings. Each description starts with the name of the building followed by "`requires`" followed by resources needed for the building in format "`<number> <name>`". Each building requires at least 1 and at most 200 units of resources.

The following p lines contain descriptions of players, each descriptions starts with the name of the player, followed by "`wants`" followed by the name of the building she wants, followed by "`has`" followed by resources that the player has in format "`<number> <name>`". Each player has at most 200 units of resources. If the player has nothing, the "`has ...`" part is omitted. You are player number 1.

All names consist of lowercase and uppercase letters and are case sensitive. Items are separated by commas and/or spaces.

Output

If it is impossible to build the building, output "`No way`". In the other case output a sequence of trade operations, one on a line, followed by building operation. See example for further clarification.

Examples

standard input
3 5 3 sheep brick wood ore wheat settlement requires 1 sheep, 1 brick, 1 wood, 1 wheat road requires 1 brick, 1 wood city requires 3 ore, 2 wheat Andrew wants settlement, has 2 sheep, 2 wood Ann wants road, has 3 brick, 2 wheat Jane wants settlement, has 3 brick, 1 wheat, 3 wood
standard output
trade with Ann wood for wheat trade with Jane sheep for brick build settlement

standard input
3 5 3 sheep brick wood ore wheat settlement requires 1 sheep, 1 brick, 1 wood, 1 wheat road requires 1 brick, 1 wood city requires 3 ore, 2 wheat Andrew wants settlement, has 2 sheep, 2 wood Ann wants road, has 3 brick, 2 wheat Jane wants settlement, has 1 sheep, 2 wheat, 3 wood
standard output
No way

standard input
5 5 3 sheep brick wood ore wheat settlement requires 1 sheep, 1 brick, 1 wood, 1 wheat road requires 1 brick, 1 wood city requires 3 ore, 2 wheat Andrew wants settlement, has 1 sheep, 3 wood Ann wants road, has 3 brick Alice wants road, has 3 brick Jane wants settlement, has 1 sheep, 2 wheat, 1 wood Joan wants road
standard output
trade with Ann wood for brick trade with Alice wood for brick trade with Jane brick for wheat build settlement

Problem H. Bilingual

Time limit: 2 seconds
Memory limit: 256 megabytes

Elliot's parents speak French and English to him at home. He has heard a lot of words, but it isn't always clear to him which word comes from which language! Elliot knows one sentence that he's sure is English and one sentence that he's sure is French, and some other sentences that could be either English or French. If a word appears in an English sentence, it must be a word in English. If a word appears in a French sentence, it must be a word in French.

Considering all the sentences that Elliot has heard, what is the minimum possible number of words that he's heard that must be words in both English and French?

Input

The first line of the input gives the number of test cases, T ($1 \le T \le 25$). T test cases follow. Each starts with a single line containing an integer N ($1 \le N \le 200$). N lines follow, each of which contains a series of space-separated "words". Each "word"is made up only of lowercase characters `a-z`. The first of those N lines is a "sentence"in English, and the second is a "sentence"in French. The rest could be "sentences"in either English or French. (Note that the "words"and "sentences"are not guaranteed to be valid in any real language.)

Each word will contain no more than 10 characters. The two "known"sentences will contain no more than 1000 words each. The "unknown"sentences will contain no more than 10 words each.

Output

For each test case, output one line containing minimum number of words that Elliot has heard that must be words in both English and French.

Example

standard input
4 2 he loves to eat baguettes il aime manger des baguettes 4 a b c d e f g h i j a b c i j f g h d e 4 he drove into a cul de sac elle a conduit sa voiture il a conduit dans un cul de sac il mange pendant que il conduit sa voiture 6 adieu joie de vivre je ne regrette rien adieu joie de vivre je ne regrette rien a b c d e f g h i j a b c i j f g h d e
standard output
1 4 3 8

Problem I. Reactor Cooling

Time limit:	1 second
Memory limit:	256 megabytes

The terrorist group leaded by a well known international terrorist Ben Bladen is buliding a nuclear reactor to produce plutonium for the nuclear bomb they are planning to create. Being the wicked computer genius of this group, you are responsible for developing the cooling system for the reactor.

The cooling system of the reactor consists of the number of pipes that special cooling liquid flows by. Pipes are connected at special points, called nodes, each pipe has the starting node and the end point. The liquid must flow by the pipe from its start point to its end point and not in the opposite direction.

Let the nodes be numbered from 1 to N. The cooling system must be designed so that the liquid is circulating by the pipes and the amount of the liquid coming to each node (in the unit of time) is equal to the amount of liquid leaving the node. That is, if we designate the amount of liquid going by the pipe from i-th node to j-th as f_{ij}, (put $f_{ij} = 0$ if there is no pipe from node i to node j), for each i the following condition must hold:

$$\sum_{j=1}^{N} f_{ij} = \sum_{j=1}^{N} f_{ji}$$

Each pipe has some finite capacity, therefore for each i and j connected by the pipe must be $f_{ij} \leq c_{ij}$ where c_{ij} is the capacity of the pipe. To provide sufficient cooling, the amount of the liquid flowing by the pipe going from i-th to j-th nodes must be at least l_{ij}, thus it must be $f_{ij} \geq l_{ij}$.

Given c_{ij} and l_{ij} for all pipes, find the amount f_{ij}, satisfying the conditions specified above.

Input

The first line of the input file contains the number N $(1 \leq N \leq 200)$ - the number of nodes and and M — the number of pipes. The following M lines contain four integer number each - i, j, l_{ij} and c_{ij} each. There is at most one pipe connecting any two nodes and $0 \leq l_{ij} \leq c_{ij} \leq 10^5$ for all pipes. No pipe connects a node to itself. If there is a pipe from i-th node to j-th, there is no pipe from j-th node to i-th.

Output

The first line of the output file must contain "`YES`" if there is the way to carry out reactor cooling, or "`NO`" if there is none. In the first case M integers must follow, k-th number being the amount of liquid flowing by the k-th pipe. Pipes are numbered as they are given in the input file.

Examples

standard input	standard output
4 6 1 2 1 2 2 3 1 2 3 4 1 2 4 1 1 2 1 3 1 2 4 2 1 2	NO
4 6 1 2 1 3 2 3 1 3 3 4 1 3 4 1 1 3 1 3 1 3 4 2 1 3	YES 1 2 3 2 1 1

Problem J. Cheese Factory

Time limit:	3 seconds
Memory limit:	256 megabytes

There are m mice living at cheese factory in Flatland. They love cheese and often eat cheese that is supposed to be sent to the cheese shop.

The i-th mouse has its cheese eating speed s_i, that means that the mouse eats s_i grams of cheese per hour.

Recently mice has learned about the plan of the cheese manufacturing. It is planned to create n cheese pieces. For each piece the following parameters are known: r_i — the hour it will be created, d_i — the hour it starts getting spoiled, p_i — its weight.

Mice have decided to each the all the manufactured cheese. Each moment each mouse can eat some cheese piece. Two mice never eat the same piece of cheese. Any moment a mouse can stop eating its piece, and can also switch to another cheese piece (but if there was another mouse that was eating that piece, it must stop).

Mice don't like eating the spoiled cheese. But they also don't like leaving cheese uneaten. So they are planning to organize eating cheese in such way that the value t such that there is some piece of cheese still eaten t hours after it started getting spoiled was minimum possible.

Help mice find out how to eat the cheese.

Input

The first line of input contains two integers n and m ($1 \le n \le 30$, $1 \le m \le 30$). The following n lines contain three integers each: p_i, r_i and d_i ($1 \le p_i \le 10^5$, $0 \le r_i < d_i \le 10^7$).

The following m lines contain one integer s_j each ($1 \le s_j \le 10^5$).

Output

Output one floating point number t. Your answer must have absolute or relative error of at most 10^{-4}.

Example

standard input	standard output
2 2 13 0 4 10 1 3 4 2	0.5

Note

The optimal behavior of mice in the first example is the following. Mouse 1 starts eating the first piece of cheese. When the second piece is manufactured it stops eating the first one and starts eating the second piece (at this moment there are 9 grams of the first piece left). Mouse 2 starts eating the first piece of cheese. After 2.5 hours mouse 1 finishes the second piece of cheese (0.5 hours after it started getting spoiled), and comes back to the first piece of cheese (mouse 2 managed to eat 5 grams of cheese from it, so its size is now 4 grams). One hour later mouse 1 finishes the first cheese piece, again 0.5 hours after it started getting spoiled.