Problem A. Exploring Pyramids

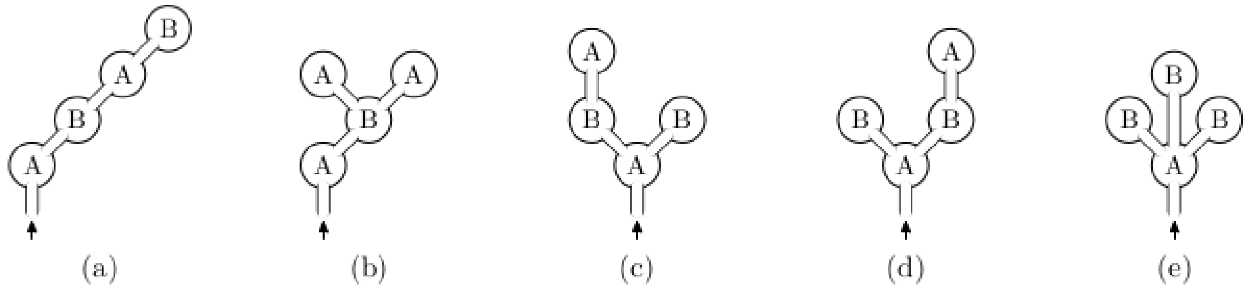
Input file:	`exploring.in`
Output file:	`exploring.out`
Time limit:	1 second
Memory limit:	512 mebibytes

Archaeologists have discovered a new set of hidden caves in one of the Egyptian pyramids. The decryption of ancient hieroglyphs on the walls nearby showed that the caves structure is as follows. There are n caves in a pyramid, connected by narrow passages, one of the caves is connected by a passage to the outer world. The system of the passages is organized in such a way, that there is exactly one way to get from outside to each cave along passages. All caves are located in the basement of the pyramid, so we can consider them being located in the same plane. Passages do not intersect. Each cave has its walls colored in one of several various colors.

The scientists have decided to create a more detailed description of the caves, so they decided to use an exploring robot. The robot they are planning to use has two types of memory — the output tape, which is used for writing down the description of the caves, and the operating memory organized as a stack. The robot first enters the cave connected to the outer world along the passage. When it travels along any passage for the first time, it puts its description on the top of its stack. When the robot enters any cave, it prints the color of its walls to its output tape. After that it chooses the leftmost passage among those that it has not yet travelled and goes along it. If there is no such passage, the robot takes the passage description from the top of its stack and travels along it in the reverse direction. The robot's task is over when it returns to the outside of the pyramid. It is easy to see that during its trip the robot visits each cave at least once and travels along each passage exactly once in each direction.

The scientists have sent the robot to its mission. After it returned they started to study the output tape. What a great disappointment they have had after they have understood that the output tape does not describe the cave system uniquely. Now they have a new problem — they want to know how many different cave systems could have produced the output tape they have. Help them to find that out.

Since the requested number can be quite large, you should output it modulo 1 000 000 000. Please note, that the absolute locations of the caves are not important, but their relative locations are important, so the caves (c) and (d) on the picture below are considered different.



Input

The input file contains the output tape that the archaeologists have. The output tape is the sequence of colors of caves in order the robot visited them. The colors are denoted by capital letters of the English alphabet. The length of the tape does not exceed 300 characters.

Output

Output one integer number — the number of different cave systems (modulo 10^9) that could produce the output tape.

Examples

exploring.in	exploring.out
ABABABA	5
AB	0

Problem B. King's Quest

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Once upon a time there lived a king and he had N sons. And there were N beautiful girls in the kingdom and the king knew about each of his sons which of those girls he did like. The sons of the king were young and light-headed, so it was possible for one son to like several girls.

So the king asked his wizard to find for each of his sons the girl he liked, so that he could marry her. And the king's wizard did it — for each son the girl that he could marry was chosen, so that he liked this girl and, of course, each beautiful girl had to marry only one of the king's sons.

However, the king looked at the list and said: "I like the list you have made, but I am not completely satisfied. For each son I would like to know all the girls that he can marry. Of course, after he marries any of those girls, for each other son you must still be able to choose the girl he likes to marry."

The problem the king wanted the wizard to solve had become too hard for him. You must save wizard's head by solving this problem.

Input

The first line of the input file contains N — the number of king's sons ($1 \leq N \leq 2000$). Next N lines for each of king's sons contain the list of the girls he likes: first K_i — the number of those girls, and then K_i different integer numbers, ranging from 1 to N denoting the girls. The sum of all K_i does not exceed 200 000.

The last line of the input file contains the original list the wizard had made — N different integer numbers: for each son the number of the girl he would marry in compliance with this list. It is guaranteed that the list is correct, that is, each son likes the girl he must marry according to this list.

Output

Output N lines. For each king's son first print L_i — the number of different girls he likes and can marry so that after his marriage it is possible to marry each of the other king's sons. After that print L_i different integer numbers denoting those girls, in arbitrary order.

Example

*

standard input	standard output

4
2 1 2
2 1 2
2 2 3
2 3 4
1 2 3 4
	2 1 2
	2 1 2
	1 3
	1 4

Problem C. Bring Them There

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	512 mebibytes

By the year 3141, the human civilization has spread all over the galaxy. The special hypertunnels are used to travel from one star system to another. To use the hypertunnel, you fly to a special location near the source star using your spaceship, activate the hyperjumper, fly through the hypertunnel, get out near your destination star and fly to the planet you need. The whole process takes exactly one day. A small drawback of the system is that for each tunnel every day only one spaceship can travel using this tunnel.

You are working in the transportation department of the "Intergalaxy Business Machines" company. This morning your boss has assigned a new task to you. To run the programming contest IBM needs to deliver K supercomputers from Earth where the company headquarters are located to the planet Eisiem. Since supercomputers are very large, one needs the whole spaceship to carry each supercomputer. You are asked to find a plan to deliver the supercomputers that takes as few days as possible. Since IBM is a very powerful corporation, you may assume that any time you need some tunnel for hyperjump, it is at your service. However, you still can use each tunnel only once a day.

Input

The first line of the input file contains N — the number of star systems in the galaxy, M — the number of tunnels, K — the number of supercomputers to be delivered, S — the number of the solar system (the system where planet Earth is) and T — the number of the star system where planet Eisiem is ($2 \le N \le 50$, $1 \le M \le 200$, $1 \le K \le 50$, $1 \le S, T \le N$, $S \ne T$).

Next M lines contain two different integer numbers each and describe tunnels. For each tunnel the numbers of star systems that it connects are given. The tunnel can be traveled in both directions, but remember that each day only one ship can travel through it, in particular, two ships cannot simultaneously travel through the same tunnel in opposite directions. No tunnel connects a star to itself and any two stars are connected by at most one tunnel.

Output

On the first line of the output file print L — the fewest number of days needed to deliver K supercomputers from star system S to star system T using hypertunnels. Next L lines must describe the process. Each line must start with C_i — the number of ships that travel from one system to another this day. C_i pairs of integer numbers must follow, pair A, B means that the ship number A travels from its current star system to star system B.

It is guaranteed that there is a way to travel from star system S to star system T.

Example

*

standard input	standard output
`6 7 4  1 6`	`4`
`1 2`	`2  1 2  2 4`
`2 3`	`3  1 3  2 6  3 4`
`3 5`	`3  1 5  3 6  4 4`
`5 6`	`2  1 6  4 6`
`1 4`	
`4 6`	
`4 3`	

Problem D. Important Roads

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

The city where Georgie lives has n junctions some of which are connected by bidirectional roads.

Every day Georgie drives from his home to work and back. But the roads in the city where Georgie lives are very bad, so they are very often closed for repair. Georgie noticed that when some roads are closed he still can get from home to work in the same time as if all roads were available.

But there are such roads that if they are closed for repair the time Georgie needs to get from home to work increases, and sometimes Georgie even cannot get to work by a car any more. Georgie calls such roads *important*.

Help Georgie to find all important roads in the city.

Input

The first line of the input file contains n and m — the number of junctions and roads in the city where Georgie lives, respectively ($2 \le n \le 20\,000$, $1 \le m \le 100\,000$). Georgie lives at the junction 1 and works at the junction n.

The following m lines contain information about roads. Each road is specified by the junctions it connects and the time Georgie needs to drive along it. The time to drive along the road is positive and doesn't exceed 100 000. There can be several roads between a pair of junctions, but no road connects a junction to itself.

It is guaranteed that if all roads are available, Georgie can get from home to work.

Output

Output l — the number of important roads — at the first line of the output file. The second line must contain l numbers, the numbers of important roads. Roads are numbered from 1 to m as they are given in the input file.

Example

standard input	standard output

6 7
1 2 1
2 3 1
2 5 3
1 3 2
3 5 1
2 4 1
5 6 2
	2
	5 7

Problem E. Network

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

In some computer company, Mouse Inc., there is very complicated network structure. There are a lot of branches in different countries, so the only way to communicate with each other is the Internet. And it's worth to say that interaction is the key to the popularity and success of the Mouse Inc.

The CEO of this company is interested now to figure out whether there is a way to attack and devastate whole structure. Only two hackers are capable to perpetrate such an outrage — Vasya and Petya, who can destroy any two channels. If after that there are at least two servers without connection between them, then they succeed.

In other words, the company is a set of servers, some of them connected with bidirectional channels. It's guaranteed that all the servers are connected directly or indirectly. The hackers' goal is to divide network into at least two parts without any connection between them. Each hacker can destroy exactly one channel. And they can't destroy the same channel together. You are asked to count the number of ways for hackers to win.

Input

There are two integer numbers (n, m) in the first line of input file: the number of servers and channels respectively ($1 \le n \le 2\,000$, $0 \le m \le 100\,000$). In the each of the next m lines there are exactly two numbers — the indices of servers connected by channel. Channels can connect a server to itself. There can be multiple channels between one pair of servers. The servers are numbered from 1 to n.

Output

There must be exactly one integer — the answer to the question described in the problem.

Example

standard input	standard output

3 3
1 2
2 3
3 1
	3

Problem F. Protoss Defense

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 mebibytes

General High Templar wants to relieve The World of disgusting invaders. His right hand, military commander Tryt o'Kill is ready to perform the plan of the destruction of the enemies. Plan is as simple as brilliant... To locate and to annihilate all enemy military targets!

So, the military commander Tryt o'Kill has been ordered to destroy enemy creatures. He has decided to provide missile attack on the enemy targets. But this task is not so simple... These targets are heavily defended by K different layers of defense. The first layer consists of *forward air defense sites* (FADS), the second layer consists of *band surface to air missiles* (BSAM), the third layer consists of *airborne interceptors* (AI), the other layers consists of different kinds of *terminal surface to air missiles* (TSAM); a $(K + 1)$-th layer contains the military targets themselves. Of course, some targets might not be defended while some others might have all defense layers.

Tryt o'Kill is good at math and he has denoted by S the set of all defense sites and by T the set of the enemy targets. A defense site might also provide protection to other defense sites in higher-numbered layers. Some defense sites might protect neither targets nor other defense sites. Commander also denoted by D_i the set of defense sites that protect the target or defense site $i \in T \cup S$. Let L_i denotes the layer's number of $i \in T \cup S$. Of course,

$$\forall\, i \in T \cup S \quad \forall\, j \in D_i \quad \Rightarrow \quad L_i > L_j.$$

Based on his past experience, Tryt o'Kill feels that while it might be possible for missiles to pass through all the defenses and reach the military targets, the probability of such a "leakage" is quite small. Instead, he believes that to destroy a target, he must first destroy all the defense sites that protect it. Therefore, he must destroy defense sites as well as targets. Destroying the i-th target or defense site has a certain military benefit and some loss. So, Commander knows the so called "cost-effectiveness" c_i of destroying the i-th target or defense site. The military commander wants to identify a set of targets and defense sites with the largest possible total "cost-effectiveness." It might be possible that providing missile attack is unreasonable and it would be better to choose nothing.

Input

Let's suppose that all targets and defense sites are numbered from 1 to n, where $n = |S \cup T|$. In the first line of the input file integer numbers n, m and K are written. The second line contains c_i $(i = 1, 2, \ldots, n)$; moreover, $-10^6 \leqslant c_i \leqslant 10^6$. The third line is made up of L_j $(j = 1, 2, \ldots, n)$ divided by space $(1 \leqslant L_j \leqslant K + 1)$. Next m lines consist of pairs of numbers i and j; each pair (i, j) means that $j \in D_i$ and $L_i > L_j$. Look at the example below to get more evident information. Keep in mind that $0 < n \leqslant 200$, $0 \leqslant K < 200$, $0 \leqslant m \leqslant 40\,000$.

Output

Write the largest possible total "cost-effectiveness" of chosen targets or defense sites in the first line of the output file. Write on the second line the number of targets of defense sites to destroy. On the third line write numbers of chosen targets or defense sites in ascending order. If there are several solutions, output any one of them.

Examples

standard input	standard output
5 4 2	6
5 -3 -4 3 2	4
3 2 2 1 1	1 3 4 5
2 4	
1 3	
3 4	
3 5	
5 4 2	0
1 2 1 -5 -5	0
3 2 2 1 1	
2 4	
1 3	
3 4	
3 5	

Problem G. Ringworld

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

The world is actually neither a disc or a sphere. It is a ring! There are m cities there, conveniently called $0, 1, 2, \ldots, m-1$, and arranged on the ring in the natural order: first 0, then 1, then 2, etc, then $m-1$, and then again 0 (as the world is a ring, remember?). You are given a collection of contiguous ranges of cities. Each of them starts at some city x, and contains also cities $x+1$, $x+2$, …, $y-1$, y, for some city y. Note that the range can wrap around, for instance if $m = 5$, then $[3, 4, 0]$ is a valid range, and so are $[1]$, $[2, 3, 4]$, or even $[3, 4, 0, 1, 2]$. Your task is to choose a single city inside each range so that no city is chosen twice for two different ranges.

Input

The input consists of several lines. The first line contains $1 \le T \le 20$, the number of test cases. Each test case consists of a number of lines. The first line contains two integers $1 \le m \le 10^9$ and $1 \le n \le 10^5$ denoting the number of cities and the number of requests, respectively. The next n lines define the ranges: the i-th row contains two integers $0 \le x_i, y_i < m$ describing the i-th range $[x_i, (x_i+1) \mod m, \ldots, y_i]$

Output

For each test case, output one line containing “YES” if it is possible to assign a unique city to each request, and “NO” otherwise.

Example

standard input	standard output

4
3 3
0 1
1 2
2 0
200000 3
100000 100000
100001 100001
100000 100001
6 6
0 1
1 2
2 3
3 4
4 5
5 0
6 6
0 0
1 2
2 3
4 4
4 5
5 0
	YES
	NO
	YES
	NO

Problem H. Hard Life

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 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

4 0	1 1

Note, that in the last example any team has hardness factor of zero, and any non-empty list of people is a valid answer.

Problem I. Inspection

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 mebibytes

You are in charge of a team that inspects a new ski resort. A ski resort is situated on several mountains and consists of a number of slopes. Slopes are connected with each other, forking and joining. A map of the ski resort is represented as an acyclic directed graph. Nodes of the graph represent different points in ski resort and edges of the graph represent slopes between the points, with the direction of edges going downwards.

Your team has to inspect each slope of the ski resort. Ski lifts on this resort are not open yet, but you have a helicopter. In one flight the helicopter can drop one person into any point of the resort. From the drop off point the person can ski down the slopes, inspecting each slope as they ski. It is fine to inspect the same slope multiple times, but you have to minimize the usage of the helicopter. So, you have to figure out how to inspect all the slopes with the fewest number of helicopter flights.

Input

The first line of the input file contains a single integer number n ($2 \le n \le 100$) — the number of points in the ski resort. The following n lines of the input file describe each point of the ski resort numbered from 1 to n. Each line starts with a single integer number m_i ($0 \le m_i < n$ for i from 1 to n) and is followed by m_i integer numbers a_{ij} separated by spaces. All a_{ij} are distinct for each i and each a_{ij} ($1 \le a_{ij} \le n$, $a_{ij} \ne i$) represents a slope going downwards from point i to point a_{ij}. Each point in the resort has at least one slope connected to it.

Output

On the first line of the output file write a single integer number k — the minimal number of helicopter flights that are needed to inspect all slopes. Then write k lines that describe inspection routes for each helicopter flight. Each route shall start with single integer number from 1 to n — the number of the drop off point for the helicopter flight, followed by the numbers of points that will be visited during inspection in the corresponding order as the slopes are inspected going downwards. Numbers on a line shall be separated by spaces. You can write routes in any order.

Example

standard input	standard output

8
1 3
1 7
2 4 5
1 8
1 8
0
2 6 5
0
	4
	1 3 4 8
	1 3 5 8
	2 7 6
	7 5

Problem J. Reserve

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 mebibytes

Protoss prepare a new attack. They have a lot of battle units to attack, but it is necessary to leave some Zealots (it is one of the most useful Protoss battle unit) on the base (as a reserve) to be sure that the base is protected from any counterattack...

One Zealot requires c_{ij} minerals to defend the base if employed from day i to day j. Attack operation proceeds for n days. Protoss commander Tryt o'Kill calculated that in day k he needs at least b_k Zealots on the base ($k = 1, 2, \ldots, n$).

The general can keep exactly b_k units on the base and pay them required number of minerals, but he wants to minimize the amount of minerals spent during all n days. The general can also keep more than b_k units on day k if this will result in smaller total amount of minerals.

Input

First line of the input file contains number of days n, $1 \leqslant n \leqslant 50$. Next n lines describe cost of keeping Zealots. So, j-th number in the $(i + 1)$-th line of the input file is $c_{i,i+j-1}$. All costs are non-negative and don't exceed 10 000. The last line contains b_k ($k = 1, 2, \ldots, n$). Consider $0 \leqslant b_k \leqslant 2000$.

All numbers in the input file are integral.

Output

In the first line write total amount of minerals General has to pay to Zealots to defend the base. Output a triangular table: j-th number of the $(i + 1)$-th line should contain number of Zealots which have to defend the base from day i to day $i + j - 1$. This number shouldn't exceed 10^9. Keep in mind that Protoss General wants to minimize amount of minerals paid to Zealots.

It is guaranteed that the optimal amount of minerals doesn't exceed 10^9.

Example

standard input	standard output

3
3 2 3
3 2
3
2 3 2
	7
	0 1 1
	0 1
	0

Problem K. Around the track

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

After The Stig's identity was revealed, the TV show Top Gear is in dire need of a new, tame racing driver to replace him. And of course you have been asked to take the job. However, you are not very fond of driving quickly, and especially not around the twisting and turning tracks they use in the show. To help you alleviate this problem, one of your algorithmic friends has suggested that you should calculate the roundtrip with the least possible amount of turning required.

The driving track consists of unique, straight lines, and there are always exactly 2 or 4 roads heading out from each node. A roundtrip must be an Eulerian circuit, i.e. it must traverse each edge of the graph exactly once, and end up where it started. (Such a circuit is guaranteed to exist in the input graphs.) Thus the total amount of turning is the sum of the turning required at each node, where continuing in a straight line means a turn of 0. The roads on the track can be driven in any direction.

Input

One line with $3 \le N \le 10^4$ — the number of nodes — and $N \le M \le 2N$ — the number of edges.

N lines with the x and y coordinates of each node, in order. $0 \le x, y \le 10^4$. The nodes have unique coordinate pairs. M lines with two space separated numbers i and j, denoting an edge between nodes i and j. The nodes are 0-indexed.

Output

The least amount of turning you must do to complete an Eulerian circuit, in radians. Absolute or relative error of 10^{-6} or less will be tolerated.

Examples

standard input	standard output

3 3
0 0
0 1
1 0
0 1
0 2
1 2
	6.283185

12 19
2 0
0 3
1 7
8 10
8 5
6 3
10 1
11 5
13 3
12 7
16 5
17 9
0 1
0 5
1 5
1 2
1 3
2 3
3 4
3 9
4 5
4 9
4 6
5 6
6 7
6 8
7 8
8 9
8 10
9 11
10 11
	22.428486

Problem L. Bart is A Tycoon

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 Mebibytes

Bart likes skateboarding and knows everything about it. He is going to become a tycoon and to have a skateboard producing factory.

Yet Bart thinks about the following problem. Assume he got M customers and C packs of skateboards. How to sell skateboards with maximum profit?

Input

The first line of input contains four integers N, S, C and R — number of cities, number of the city where Bart's factory is placed, number of packs of skateboards Bart got for sale and penalty for car accident ($1 \leq N,\ C \leq 100$, $1 \leq S \leq N$, $0 \leq R \leq 10^4$).

The second line contains an integer M — number of customers ($1 \leq M \leq 100$).

Next M lines consist of customer descriptions and contain three positive integers P_i, C_i and M_i each, where P_i is the customer city number ($1 \leq P_i \leq N$), C_i is the maximal number of packs of skateboards that customer could buy ($1 \leq C_i \leq 100$) and M_i is the profit that Bart gets from selling one pack to this customer ($1 \leq M_i \leq 10^4$). A customer can buy any integer number of packs of skateboards between 0 and C_i, inclusive.

Next line consists of one integer E — the number of roads ($0 \leq E \leq 10^4$). All roads between cities are directed, but car accidents are still possible.

Next E lines consist of road descriptions. Each road is described by six integers V_1, V_2, U, F, P_1 and P_2 — numbers of beginning and ending city on this road, maximum road capacity in trucks, fuel expenses for one truck to pass this road and two special parameters, related to car accident probability, respectively ($1 \leq V_1, V_2 \leq N$, $V_1 \neq V_2$, $1 \leq U \leq 100$, $1 \leq F \leq 10^4$, $0 \leq P_1 \leq P_2 \leq 100$). One truck can carry one pack of skateboards. If during the whole period of deliveries, there will be more than U trucks on the road, there would be a traffic jam and all these trucks won't arrive in time. If $U > 1$, the probability of a car accident for i-th truck on the road is $P_1 + (P_2 - P_1)(i - 1)/(U - 1)$. In case of $U = 1$, the probability of a car accident for a truck is $P_1 = P_2$.

If a truck has accident, Bart pays R as a penalty and then the truck continues delivery. A truck can have no more than one accident per a road.

Output

Output expected profit with accuracy 10^{-2}. Note that Bart need not sell all C packs of skateboards.

Example

standard input	standard output

3 1 3 100
2
2 2 1000
3 1 900
2
1 3 2 5 10 10
1 2 2 5 10 20
	2845.00

Problem M. Minimal Cut Matrix

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 megabytes

Consider a connected weighed undirected graph G. A *minimal edge cut* between vertices u and v is the set $C_{u,v}$ of edges with minimal total weight, such that each path from u to v contains at least one edge from $C_{u,v}$. Let us denote the weight of a minimal edge cut between vertices u and v as $c_{u,v}$.

The matrix $c_{u,v}$ is called the *minimal cut matrix* of graph G. You are given a matrix $c_{u,v}$. Find the graph G such that $c_{u,v}$ is the minimal cut matrix of this graph, or detect that there is no such graph.

Input

The first line of the input file contains n — the size of the matrix ($2 \le n \le 50$). The following n lines contain n integer numbers each — the entries of the matrix. It is guaranteed that the matrix is symmetric and has zeroes at the main diagonal. All other entries are positive and do not exceed 1000.

Output

If there exists a graph G such that the matrix in the input file is its minimal cut matrix, print "`YES`" at the first line of the output file. In this case print m — the number of edges in the graph at the second line, and describe edges in the following m lines. Each edge must be described by the numbers of vertices it connects, and the weight of the corresponding edge. There must be no loops, neither parallel edges. No weight must exceed 1000.

If there is no such graph, print "`NO`" at the first line of the output file.

Example

standard input	standard output

3
0 2 2
2 0 2
2 2 0
	YES
	3
	1 2 1
	1 3 1
	2 3 1