Problem A. Maximum Matching

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

Your task is to find maximum matching in the given bipartite graph.

Input

The first line contains two integers n and m ($1 \le n, m \le 250$) — number of the vertices in the part A and in the part B.

The following n lines contain description of the edges. The i-th vertex from A is described in the $(i+1)$-th line. Each line contains the vertices from B connected to the i-th vertex from A. Vertices in both parts are numbered independently. Each line ends with 0.

Output

The first line must contain integer l — the number of edges in maximum matching. Each of the following l lines must contain two integers u_j, v_j — the edges of maximum matching.

Example

standard input	standard output
2 2 1 2 0 2 0	2 1 1 2 2

Problem B. Maximum by Inclusion Matching

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

Your task is to find maximum by inclusion matching in the given bipartite graph. Maximum by inclusion matching is a matching for which it is impossible to add an edge from the given graph preserving a matching property. I.e. a maximum by inclusion matching is not included in other matching.

Input

The first line contains two integers n and m ($1 \leq n, m \leq 100000$) — the number of vertices in the part A and in the part B.

The following n lines contain description of the edges. The i-th vertex from A described in the $(i+1)$-th line. Each line contains the vertices from B connected with the i-th vertex from A. The vertices in both parts are numbered independently. Each line ends with 0.

Output

The first line must contain integer l — number of edges in maximum by inclusion matching. Each of the following l lines must contain two integers u_j, v_j — the edges of maximum matching.

Example

standard input	standard output
2 2 1 2 0 2 0	2 1 1 2 2

Problem C. Minimum Edge Cover

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

Your task is to find minimum edge cover in the given bipartite graph.

The set of edges R is an edge cover if and only if each vertex of the given graph is incident to at least one edge from R.

Input

The first line contains two integers n and m ($1 \le n, m \le 250$) — the number of vertices in the part A and the number of vertices in the part B.

The following n lines contain edges description. The i-th vertex from A is described in $(i+1)$-th line. Each line contains numbers of the vertices from B connected to the i-th vertex from A. The vertices from parts are numbered independently. Each line ends with 0.

Output

The first line must contain integer l — the number of edges in the minimum edge cover. Each of the following l lines must contain two integers u_j, v_j — edges of the minimum edge cover. If there are multiple solutions, print any of them.

Example

standard input	standard output
2 2 1 2 0 2 0	2 1 1 2 2

Problem D. Cubes

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

Parents' gift to Peter is a set of kids' cubes. Each cube has six faces and on each face there was written one letter.

Peter wants to show his older sister that he learnt to read. To make it he wants to form her name from those cubes. But it is difficult for Peter because different letters can be on the same cube and in this case Peter can not use both of them to make name of his sister. But it is possible that the same letter is written on different cubes. Your task is to help Peter.

You are given the set of cubes and the name of his sister. You need to check if it is possible to make his sister's name from the cubes given. If it is possible you need to determine the order of cubes in which Peter needs to arrange them.

Input

The first line contains integer n ($1 \le n \le 100$) — the number of the cubes.

The second line contains the name of Peter's sister — the word consisting of capital Latin letters. The length of the name does not exceed 100.

Each of the following n lines contains 6 letters (only capital Latin letters) which are written on the corresponding cube.

Output

Print "`YES`" in the first line if it is possible to form sister's name from the cubes and "`NO`" in the other case.

If the answer is "`YES`" print in the second line m different numbers from the range $1 \ldots n$, where m — the number of letters in sister's name. The i-th number must be equal to the number of cube which is needed to put in the i-th place. The cubes are numbered from 1 in the order they appear in the input. If there are multiple solutions you are allowed to print any of them.

Examples

standard input	standard output
4 ANN ANNNNN BCDEFG HIJKLM NOPQRS	NO
6 HELEN ABCDEF GHIJKL MNOPQL STUVWN EIUOZK EIUOZK	YES 2 5 3 1 4

Problem E. Language Codes

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

An important task of the football forum organizers is to create good conditions for press representatives. In order to automate the processing of requests from press, it was decided to create a system of two-letter language codes (like Internet domain names of countries).

To assign a language code, the English name of the language is divided into head (the first letter) and tail (all letters except the first). Correct code consists of two lowercase letters exactly. The first letter must be equal to the first letter of the language name and the second letter must be in the tail of the name. For example, "`en`", "`eg`" and "`eh`" are correct codes for English language, but "`uu`", "`um`" and "`kr`" are not correct codes for Ukrainian language. Of course, all codes must be unique, i.e. for any two languages the corresponding codes must be distinct.

You are given the set of languages. Calculate maximum number of languages for which it is possible to assign correct and unique codes at the same time.

Input

The input contains at most 40 tests.

The first line of each test contains integer N ($1 \le N \le 100$) — the number of languages.

Each of the following lines contains the languages consisting of uppercase and lowercase Latin letters. Length of each language is from 1 to 20.

The input ends with the test for which $N = 0$. Do not print answer for this test.

Output

Print a single line for each test in the input. Each line must contain single integer — maximum number of languages for which it is possible to assign correct and unique codes at the same time.

Examples

standard input	standard output
5 Ukrainian English Japanese Chinese Russian 0	5
4 ABC ACD ADB ABD 0	3
6 AA AAB AABC ACABABABABABD ADE AEBC 0	5

Problem F. Minimum Vertex Cover

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

You are given a graph and its maximum matching. Your task is to find minimum vertex cover in this graph.

Input

The first line contains two integers m and n ($1 \le m, n \le 4\,000$) — the sizes of the parts.

Each of the following m lines contains the list of the edges which begin with the corresponding vertex from the first part. This list begins with the integer K_i ($0 \le K_i \le n$) — the number of the edges. After that the vertices from the second part connected to the vertex from the first part are given in an arbitrary order. The total sum of all the values K_i does not exceed $500\,000$.

The last line contains some maximum matching in this graph — m integers $0 \le L_i \le n$ — the vertex from the second part matched to the i-th vertex from the first part, or `0`, if the i-th vertex from the first part is unmatched.

Output

Print in the first line the size of minimum vertex cover.

Print in the second line the integer S — the number of the vertices from the first part and after it print S integers — the vertices from the first part which belong to minimum vertex cover in ascending order.

Print in the third line the description of the vertices from the second part in the similar format.

Example

standard input	standard output
3 2 2 1 2 1 2 1 2 1 2 0	2 1 1 1 2

Problem G. Minimization of the Most Expensive Edge

Input file:	*standard input*
Output file:	*standard output*
Time limit:	6 seconds
Memory limit:	256 mebibytes

You are given a complete bipartite graph. Each part contains n vertices. For each edge (i, j) $(i, j = 1 \cdots n)$ its cost $g_{i,j}$ is given.

Your task is to find such perfect matching, in which the cost of the most expensive edge is as minimal as possible.

Input

The first line contains integer n $(1 \leq n \leq 500)$.

Each of the following n lines contains n integers — the matrix with edges cost: the j-th number in the i-th line equals the cost $g_{i,j}$ of the edge from vertex i in the first part to the vertex j in the second part. All costs are integers with absolute values not exceeding 10^6.

Output

The first line must contain the minimum possible cost of the most expensive edge.

The following n lines must contain required perfect matching. Each of n lines must contain two integers a_i, b_i — endpoints of the next edge.

Example

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

Problem H. Checking Program

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

Your task is to check if the given matching is maximum in the given graph. If it is not true you need to find the matching which is larger than the given.

Input

The first line contains two integers m and n ($1 \le m, n \le 4\,000$) — the sizes of parts.

Each of the following m lines contains the list of edges going from the corresponding vertex in the first part. This list starts with integer K_i ($0 \le K_i \le n$) — the number of the edges followed by the vertices from the second part, connected to the corresponding vertex from the first part in arbitrary order. The total sum of all values K_i does not exceed $500\,000$.

The last line contains some maximum matching in given graph — m integers $0 \le L_i \le n$, where the i-th value is the vertex from the second part matched to the i-th vertex from the first part, or `0`, if the i-th vertex from the first part is unmatched.

Output

The first line must contain word "`YES`" if the given matching is maximum and "`NO`" in the other case.

If the given matching is not maximum, the second line must contain integer l — the number of the edges in a greater matching. Each of the following l lines must contain two integers u_j, v_j — edges from the greater matching.

Examples

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

Problem I. Dominoes

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

There is a $N \times N$ square chessboard. P squares were removed from the chessboard. Find out if it is possible to cover the remaining part of the chessboard with domino pieces (each piece is 2×1).

Put each domino piece on two neighboring cells exactly. No two pieces can cover the same cell.

Your task is to find the required tiling, if it exists.

Input

The first line contains two integer numbers N $(1 \leq N \leq 40)$ and P $(0 \leq P < N^2)$ separated by a space.

Each of the following P lines contains a pair of numbers separated by a space — coordinates of the removed cell $(1 \leq X_i, Y_i \leq N)$. The bottom left square has the coordinates $(1, 1)$, the bottom right square — $(N, 1)$.

Output

If the required covering exists, output "`Yes`" in the first line and "`No`" in the opposite case.

If the first answer is positive, then output in the second line integer number N_h — the number of horizontally placed pieces. Each of the following N_h lines should contain two integers — the coordinates of the left cell covered by a corresponding piece.

Output in the next line N_v — the number of vertically placed pieces. And the following N_v lines should contain the coordinates of the bottom cell covered by a corresponding piece.

Example

standard input	standard output
4 10 1 3 1 2 1 1 2 1 3 1 4 1 3 2 4 2 3 3 4 3	Yes 2 1 4 3 4 1 2 2

Problem J. Minimum Path Cover

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

You are given a directed acyclic graph (DAG). Your task is to find minimum path cover of this graph, i.e. to find minimum set of paths such that each of the vertices belongs to exactly one path. Paths can have lengths than equal to zero (i.e. a path can consist of only one vertex).

Input

The first line contains two integers n and m — the number of vertices and edges in the graph ($1 \le n \le 100$).

Each of the following m lines contains two integers — description of the edges.

It is guaranteed that graph is acyclic and it does not contain loops and multiple edges.

Output

Print in the first line the minimum number of paths. Then print each path one per line. If there are multiple solutions you are allowed to print any of them.

Examples

standard input	standard output
4 4 1 2 2 4 1 3 3 4	2 1 2 4 3
7 8 1 2 1 3 2 4 3 4 4 5 4 6 5 7 6 7	3 1 2 4 5 7 3 6

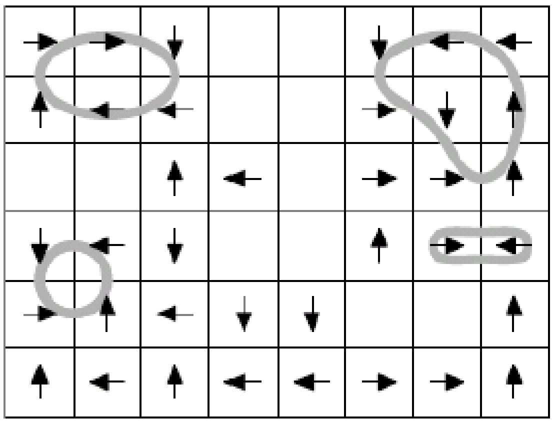
Problem K. Arrows

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

Kostya found a puzzle game which his grandfather played being a child — a board containing $n \times m$ cells. Rotating arrows were attached to some of the cells (Kostya thinks that many years ago the arrows were on each cell). Unfortunately, Kostya's grandfather did not remember what had to be done in this puzzle, so Kostya invents rules by himself.

He noted the following feature. Let's direct each arrow to one of the side neighboring cells, which also has an arrow (we consider that each arrow on the board has the neighboring cell with an arrow). Now there are cycles on the board and we move on these cycles in directions of the arrows.

For example, the picture shows 4 cycles.



Kostya invented two tasks for this puzzle: "rotate the arrows so that the number of cycles becomes minimum possible" and "rotate the arrows so that the number of cycles becomes maximum possible". Your task is to find these two numbers.

Input

The first line contains two integers n and m ($1 \leq n, m \leq 100$).

Each of the following n lines contains m symbols — the description of the board. The symbol "`?`" corresponds to the cell with an arrow and the symbol "`.`" — to the empty cell.

Output

Print the board which contains minimum possible number of cycles and print the board which contains maximum possible number of cycles. Separate both boards by one empty line.

To print the directions of the arrows change symbols "`?`" to letters which correspond to directions. Use letters "`R`", "`L`", "`U`" and "`D`" to right, left, up and down directions respectively.

Example

standard input	standard output
3 4 ???? ???? .??.	RDLD ULUL .RU. DRLD UDDU .UU.

Problem L. The Greatest Dominance

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

You are given a bipartite graph. The first part contains N vertices and the second part contains M vertices.

Let's choose some subset of the vertices from the first part and let's call them X. Let Y be such a subset of the vertices from the second part that are connected to at least one vertex from X (i.e. Y is a subset of the neighboring vertices of X).

Your task is to find such subset X making the expression $|X| - |Y|$ as maximal as possible.

Input

The first line contains two integers N and M ($1 \le M, N \le 300$).

The following N lines contain description of the vertices from the first part. Each description is given in the format: the i-th line begins with the number D_i equals to the numbers of neighbors of vertex i from the first part. After it D_i integers are given — the numbers of neighbors in the second part of the vertex i from the first part.

Output

Print in the first line the number of vertices in X.

Print in the second line the vertex numbers from X.

It is possible that X is empty. In this case the expected expression value equals to 0.

Examples

standard input	standard output
1 3 3 1 2 3	0
3 2 1 1 1 1 2 1 2	2 1 2

Problem M. Exhibition

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

An exhibition of the arithmetical progressions will be open in Berland soon. Each of N presented progressions is characterized by two integer numbers A_i and B_i. These numbers mean that all numbers X belong to the i-th progression if $X = A_i \cdot K + B_i$ for some non-negative integer K. Organizers decided that for each sequence a unique number from 1 to M is to be assigned. The organizers want as much as possible progressions to be assigned under such a number, that would be an element of this progression. They hired you to do this job.

Input

The first line of the input contains two integer numbers N and M ($1 \le N \le 300$; $N \le M \le 10^9$). Each of the following N lines contains a pair of numbers A_i and B_i ($-10^9 \le A_i, B_i \le 10^9$).

Output

Print in the output the required assignment of numbers to the sequences — N numbers separated by spaces. All numbers should be unique. If there are several solutions, output any of them.

Example

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

Problem N. Voting Block

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

The members of a committee vote "`Yes`" or "`No`" on various issues. However, some pairs of committee members have formed alliances, promising never to cast opposite votes. No one is willing to vote contrary to her own opinion, so some of the members may need to abstain from voting in order to avoid conflict between allies.

We have devised a method to determine who should abstain. Before each issue is voted on, we randomly assign each committee member an identifying number 1, 2, $\cdots$, n. The member will then indicate her opinion on the issue. Then we will calculate the smallest collection of abstentions that will avoid conflict.

You are given the descriptions of the voters, calculate the smallest collection of abstentions that will avoid conflict.

Input

The first input line contains the number n $(1 \le n \le 50)$ – the number of voters. The i-th following line contains the description of the i-th voter. The description of the i-th voter contains its member's opinion, either "`Y`" or "`N`", followed by the identifying number of each higher-numbered member who has formed an alliance with its member (or nothing if there is no such member).

Output

In the first line print the minimal amount of abstentions. In the second line print their indexes in ascending order, if there is more than one answers, print the one that is earliest lexicographically.

Examples

standard input	standard output
2 Y 2 N	1 1
3 N 2 N 3 Y	1 2

Problem O. Chessboard

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

Bob likes to play unusual chess. Once, when his brother Nick was very small, Bob got back from school and saw that his lovely chessboard has been repainted by Nick. Bob was not very angry because he was very fond of his brother. Since Nick had only black and white colors, each cell of the board has been painted in one of these two colors.

Bob decided to correct brother's mistake and paint the board so that it becomes a chessboard again. In other words Bob wants checkboard to have only black cells neighbouring white cells and vice versa.

Bob wants to paint only the entire diagonals of the board. Bob decided not to spend a lot of time so he is interested in such a way of painting, which contains the least number of actions. In one action Bob can paint some whole diagonal of the board in black or white color. There are diagonals of two types. Diagonal, which lies on the line, directed to the left and down, is a diagonal of the first type. Diagonal, which lies on the line, directed to the right and down, is a diagonal of the second type.

Your task is to help Bob. You are given the spoiled by Nick chessboard. You need to determine minimum number of actions after which Bob can repaint the board in such a way that it becomes a chessboard.

Input

The first line contains two integers n and m ($1 \le n, m \le 100$) — the number of rows and the number of columns of the board.

The following n lines contain description of the board. Each line contains m symbols and describes rows of the board. Letter 'W' corresponds to a white cell of the board and letter B — to a black cell of the board.

Output

Print in the first line integer p — the minimum number of operations (diagonals) needed to be repainted.

Print in the following p lines description of the painting. Each line must contain: the type of diagonal (1 or 2), coordinates of some cell painted on this diagonal (the number of row and column of this cell) and color in which this diagonal is painted (letter 'W' for white color and 'B' for black).

Examples

standard input	standard output
3 3 WBB BWB BBW	1 1 3 1 W
3 3 WBW WWB WWW	1 2 3 2 B
1 3 WWW	1 1 1 2 B

Problem P. Kind Vitya

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

In the summer camp teachers are able to hold N different lectures. Unfortunately the camp will be conducted during only M days.

For humanitarian reasons at most one lecture can be hold in one day. Teachers will be in the camp not for all the time, that's why some lectures can be held only on certain days only. It was decided to hold the maximum number of lectures, because the national team must prepare for the international competition as good as possible.

Kind Vitya gave to the participants the information about days when each lecture can be held. Participants do not know which of the camp schedule will be chosen, but they are interested in lectures which will be held in any schedule.

Input

The first line contains two integers — N and M ($1 \le N, M \le 500$).

Each of the following lines contains description of the lectures — one per line. For each lecture the numbers of days within this lecture can be held is specified. The numbers in line are separated by one or more spaces. The days and the lectures are numbered from one.

Output

Print the lecture numbers which will be held in any camp schedule in ascending order.

Example

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

Problem Q. Edges Belonging to Some Maximum Matching

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

You are given a bipartite graph. Your task is to find edges which belong to at least one maximum matching in this graph.

Input

The first line contains three integers n, m, k ($1 \le n, m \le 500$, $0 \le k \le min(n \cdot m, 10^5)$), where n, m are the sizes of the parts and k is the number of the edges in this graph.

Each of the following k lines contains two integers a_i, b_i: descriptions of the endpoints of an edge. The value a_i ($1 \le a_i \le n$) equals to the vertex number from the first part and b_i ($1 \le b_i \le m$) equals to the vertex number from the second part.

Output

Print in the first line the required number of required edges.

Print in the second line the indexes of all edges which belong to at least one maximum matching. All indexes must be printed in ascending order. Edges numbered from one, in the same order that they appear in the input.

Example

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