Problem A. Bridges, Edges and Leaves

Input file:	`bel.in`
Output file:	`bel.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

Research Institute of Leaves and Edges (RILE) and Research Institute of Bridges (RIB) have a lot in common. They often work in tight collaboration. But today they faced a problem that lies exactly between their research fields. They need to build an undirected graph with B bridges, E edges and L leaves. After a lot of study, they invited Vasya to deal with this problem. But as you know, Vasya is specialist in strings, so he asked for your help.

Remember that, for a given undirected graph, *bridge* is an edge such that deleting it increases the number of connected components, and *leaf* is a vertex of degree 1.

Input

The input file contains three integers B, E and L such that $0 \leqslant B, E, L \leqslant 1000$.

Output

You must output the description of any graph consisting V vertices, with exactly B bridges, E edges and L leaves, or `IMPOSSIBLE` if there is no such graph. Here V is non-negative integer number which does not exceed 2000. The first line must contain V. The following E lines contain pairs of numbers u_i and v_i indicating that vertices u_i and v_i ($1 \leqslant u_i, v_i \leqslant V$) are connected by a single edge. Loops, duplicate edges and vertices with zero degree are not allowed.

Examples

bel.in	bel.out
1 4 1	4 1 2 2 3 3 4 2 4
5 4 3	IMPOSSIBLE
2 2 4	4 1 2 3 4

Problem B. Cutting the Rectangle-2

Input file:	`cutrect2.in`
Output file:	`cutrect2.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

After cutting the rectangle from some old task, Vasya is asked to cut another rectangle for Research Institute of Cutting and Embedding (RICE).

This rectangle $m \times n$ is also cut from a sheet of grid paper in such way that the cuts are done along the lines of the grid. But now, some cells of the rectangle (at least one) are important, and the other cells are not. Vasya is asked to cut this rectangle with an infinite straight line of given direction in such way that the ratio of the total important area in resulting parts is equal to given one.

Could you help him again?

Input

The input consists of no more than two thousand test cases. Each case starts with a line containing three numbers m, n and R, where m and n are integers between 1 and 100, and R is the ratio of area of the leftmost part to area of the rightmost one. The "leftmost" and "rightmost" parts are defined according to moving along given non-zero cut direction vector, which is specified in the second line of the case by two integers dx and dy. Both these numbers do not exceed 1000 by an absolute value. The rest m lines of the case contain the rectangle map — n characters in each line, where '`*`' stands for an important cell, and '`.`' stands for an unimportant one. The input is terminated by a line of three zeroes.

It is guaranteed that R is a real number between 0.001 and 1000 given as precisely as possible. The total number of cells over all test cases in a single input does not exceed $2 \cdot 10^6$.

Output

For each case, write one of the points on the line of cut as precisely as possible. Adhere to format of the sample output. The judging program will check that the ratio of important areas has an absolute error no more than 10^{-6} assuming your output to be written as precisely as possible. Lower-left corner of the rectangle has coordinates $(0, 0)$.

Examples

<table>
<tr><th>cutrect2.in</th><th>cutrect2.out</th></tr>
<tr><td><pre>1 1 1.0
-1 1
*
1 1 7.0
-1 1
*
1 1 7.0
1 -1
*
2 3 9.0
2 -3
.
.**
0 0 0</pre></td><td><pre>Cut rectangle 1 at (0.5, 0.5)
Cut rectangle 2 at (1, 0.5)
Cut rectangle 3 at (0, 0.5)
Cut rectangle 4 at (-0.011237390046160782, 2)</pre></td></tr>
</table>

Problem C. Handled graph

Input file:	`graph.in`
Output file:	`graph.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

As you know, Vasya sometimes works in other institutes. Now he was sent to RIGHT (Research Institute of Graph Handling Theory). And there he... Yes, he handles a graph. Vasya knows how to handle a tree. Of course, you know it too. But Vasya's task is much harder. He should handle a directed graph! Vasya doesn't even know what he should do...

A very clever book says that Vasya should write a number near each vertex (it is called *layer number*) in such a way that any edge starting at vertex with layer number n can go only to vertex with layer number n or $n+1$. The minimal layer number should be exactly 0; moreover, if the maximal layer number is S, all the integers between 0 and S must be used at least once. Of course, there are many ways to handle a graph (for example, put all vertices onto layer 0). But the canonical way is the one which maximizes S.

Your task is to find any of the canonical handling ways of a given graph.

Input

The first line contains two integers n and m ($1 \le n \le 300$, $0 \le m \le 100\,000$) — the number of vertices and the number of edges, respectively. The next m lines contain information about edges. Each of these lines contains two integer numbers a and b ($1 \le a, b \le n$) — numbers of starting and finishing vertices of the corresponding edge. Loops and multi-edges are allowed.

Output

Write one line with N integers; i-th of them should be the layer number of i-th vertex (vertices are numerated as in input).

Examples

graph.in	graph.out
`2 1` `1 2`	`0 1`

`2 2`
`1 2`
`2 1`
	`0 0`

Problem D. Shortest Knight's Path

Input file:	`ndist.in`
Output file:	`ndist.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

Research Institute of Stepping Knights (RISK) gives Vasya a **very difficult** problem — find the shortest knight path between two cells on an infinite chess board. Because Vasya doesn't know chess rules, he asked his friend Petr to translate this problem into pure math language. After five minutes of thinking, Petr wrote a fuzzy problem statement:

Given x_1, y_1, x_2, y_2, find $\{\Delta x_i, \Delta y_i, c_i\}_{1 \le i \le k}$ such that:

$$
\begin{aligned}
&c_i \ge 0\\
&\Delta x_i \in \{-2, -1, 1, 2\}\\
&\Delta y_i \in \{-2, -1, 1, 2\}\\
&\Delta x_i^2 + \Delta y_i^2 = 5\\
&x_1 + \sum_{1 \le i \le k} c_i \Delta x_i = x_2\\
&y_1 + \sum_{1 \le i \le k} c_i \Delta y_i = y_2\\
&l = \sum_{1 \le i \le k} c_i \to \min
\end{aligned}
$$

Input

First line contains a single integer ($1 \le T \le 10$) — number of test cases. Next T lines contain four integers x_1 y_1 x_2 y_2 each ($-10^6 \le x_1, y_1, x_2, y_2 \le 10^6$).

Output

For each test case, the output should contain $(k + 1)$ lines. First of them should contain two integers l and k ($0 \le k \le 8$) separated by a single space. Each of the next k lines should contain three integers Δx_i, Δy_i and c_i separated by single spaces. If there are several answers with the minimal possible value of l, output any of them. Output should **not** contain blank lines between test cases.

Example

ndist.in	ndist.out
2 0 0 1 0 0 0 1 1	3 3 1 -2 1 2 1 1 -2 1 1 2 2 -1 2 1 2 -1 1

Problem E. Next, Please. . .

Input file:	`next.in`
Output file:	`next.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

There was a unique device in the Research Institute of Next Generation (RING). It generates . . . well, it generated the next number partition of an integer number n in lexicographical order. If the given partition was the last partition of n, it generated the first partition of $n + 1$.

Unfortunately, somebody recently tried to investigate shockproof characteristics of the device. But after this device was thrown out from the 10th floor window, something strange happened to it. For example, at 10:02 it generated 5 from 3 2. At 10:03 it generated 6 5 from 5 3 3. The scientists of RING think that the device now considers only partitions consisting only of numbers that are not less than some integer m, for example, minutes count.

Vasya's task is to help to check this hypothesis. He needs to write a program for simulating proposed new behavior of the device. More formally, his program should consider an infinite sequence $S(m)$ of number partitions which looks like $P(m)$ $P(m+1)$ $P(m+2)$ $\ldots$, where $P(t)$ is lexicographically ordered sequence of all partitions of t consisting of numbers which are not less than m. Here a partition is defined as $t = a_1 + a_2 + \ldots a_k$, where $k \geqslant 1$, and $a_1 \geqslant a_2 \geqslant \ldots \geqslant a_k$. Lexicographical order is defined by saying that the first non-equal elements at position p of two partitions a and b (where b occurs later than a in $P(t)$) must satisfy $a_p < b_p$. For example, the start of this sequence for $m = 2$ looks as follows: 2, 3, 2 2, 4, 3 2, 5, 2 2 2, 3 3, 4 2, 6, . . .

The task of the program is, given m and an element of $S(m)$, to generate the next one.

Input

The first line of input contains two integers k and m ($1 \leqslant k \leqslant 50\,000$, $1 \leqslant m \leqslant 50\,000$). The second line consists of k integers a_i ($m \leqslant a_i \leqslant 10^6$). It is guaranteed that $a_1 \geqslant a_2 \geqslant \ldots \geqslant a_k$.

Output

The first line should contain the length of the next element k'. The second line contains k' numbers forming a non-increasing sequence — the next element. It is guaranteed that k' will never exceed $50\,000$.

Examples

next.in	next.out
2 2 3 2	1 5
3 3 5 3 3	2 6 5
1 2 5	3 2 2 2

Problem F. Cutting a Pie

Input file:	`pie.in`
Output file:	`pie.out`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

> My pies! The toggles do nothing!
>
> The cry of a cook who faced a non-working oven

Vasya worked hard for RIGS (Research Institute of Given Strings) for many years. But recently he understood that he should think about children. So he is going to get a job at RICH (Research Institute of Children's Happiness). He sent his curriculum vitae to the recruiting department, and they gave him a simple task to check his skills. Vasya isn't familiar with such problems yet, so he asked for your help.

The problem is about N children and only one pie. Each of these children has her (or his) own conception of a good piece of a pie. It can be formalized in the following way. Each girl (or boy) has two rational numbers a and b in her (or his) mind. She (or he) considers a piece of a pie of size s to be good if and only if $a \leqslant s$ and $s \leqslant b$.

The task is to divide a pie between children in such a way that each of them would think that her (or his) piece is good.

To cut a pie, the following sequence of actions should be performed:

1. Number all children by integers from 1 to N.
2. For all i, assign some integer k_i ($1 \leqslant k_i \leqslant K$) to a child with number i.
3. For all i, give to child with number i a piece of the pie with size $\frac{k_i}{\sum_{i=1}^{N} k_i}$.

Input

The first line of the input file contains two integers N ($1 \leqslant N \leqslant 300$) and K ($1 \leqslant K \leqslant 100$). Then follow N lines. The line $(i+1)$ contains two rational numbers a_i and b_i separated by exactly one space. Each of these numbers is given as a ratio of two integers. These integers lie between 0 and 10^9, inclusive. Second integer in each ratio isn't equal to 0.

Output

If there is no assignment which solves this problem, print "`No solution`". If there is a solution, print one line containing N integers — the numbers k_i which should be assigned to children. If there is more then one solution, you should choose the one with the smallest $\sum_{i=1}^{N} k_i$. If there is still ambiguity, compare sequences of numbers k_i lexicographically and choose the smallest one.

Recall that an array $a_1, a_2, \ldots, a_p$ is said to be *lexicographically smaller* than an array $b_1, b_2, \ldots, b_p$ if for the smallest i such that $a_i \neq b_i$, it is true that $a_i < b_i$.

Examples

pie.in	pie.out
1 2 0/1 1/2	No solution
3 2 0/1 1/1 0/1 1/1 0/1 1/4	1 2 1

Problem G. Primes in a Sequence

Input file:	`primes.in`
Output file:	`primes.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

R.I.P. (Research Institute of Primes) calls for help! Vasya is here today to help them to search for primes in very specific sequences. These sequences start with an arbitrary positive integer a_1. The following numbers of the sequence are evaluated using formula $a_{i+1} = d(a_i) + i$. Here $d(x)$ stands for the number of divisors of x. For example, if the sequence starts from $a_1 = 2^{32\,582\,657} - 1$, then $a_2 = 3$, $a_3 = 4$ and $a_4 = 6$.

The (l, r)-*window* of some sequence is defined as the subsequence of this sequence between its l-th and r-th element, inclusive.

Vasya's task now is to determine maximal possible number of primes (each occurence of any repeated prime is accounted) in (l, r)-window over all sequences of such kind for given l and r. The problem is too hard for him because there is an infinite number of such sequences. For example, the sequence described above is an example of optimal sequence for $(1, 4)$-window. So he asked you for help.

Input

The input consists of no more than one thousand cases. Each case is a single line with two integers l and r ($1 \leqslant l < r \leqslant 10^6$). Input is terminated by the line of two zeroes. The total sum of $l + r$'s over all cases in a single input does not exceed 10^7.

Output

Output the maximal possible number of primes as shown below.

Examples

primes.in
1 4 6 7 10 20 0 0

primes.out
Window 1: There can be up to 2 primes. Window 2: There can be up to 1 primes. Window 3: There can be up to 4 primes.