Problem A. Army of Mages

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

Everybody was silent. It became known that after the North battle all warriors of the great Coalition Army would stop defending their kingdom as battle mages and would serve as coaches. So it was the time to build a new powerful army. It was the wise Sandro who broke the silence and began to write down the names of the candidates. Finally, he wrote down $5N$ names. But only N best mages had to be chosen according to the law of the kingdom. After a long discussion it was decided to build an army in such a way that every two mages in this army would *respect* each other.

All mages in the kingdom live in their own houses, situated at the points on the plane with integer coordinates. Somewhy two mages respect each other if and only if a line segment connecting their houses contains at least one point with integer coordinates, different from endpoints of this segment. For example, if there are houses at points $(1, 1)$ and $(5, 5)$ then their inhabitants respect each other, because there is a point $(2, 2)$ on this segment. In the same time, inhabitants of the houses situated in $(0, 0)$ and $(1, 10)$ don't respect each other.

Help the government to build an army!

Input

The first line contains an integer N ($1 \leqslant N \leqslant 5000$). The i-th of the next $5N$ lines contains a pair of integers x and y, not exceeding 10 000 by their absolute values — coordinates of the house of the i-th candidate. All houses are situated at different points.

Output

If the required army can be built, output «OK» in the first line and in the second line output N space-separated numbers of the selected candidates in any order. If there are several possible answers, you can output any of them. If no army can be built, output «IMPOSSIBLE».

Example

army.in	army.out
2 1 1 5 5 0 0 2 2 0 10 6 6 7 7 8 8 9 9 10 10	OK 2 1

Problem B. Aztec Treasure

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

During the recent excavations in Teotihuacan archeologists found a strange casket, the contents of which was probably used during the legendary corbans held by Montezuma, and a lot of equal rectangular bone pieces of size 1×2.

Archeologists found out that in order to open the casket you should tile the rectangular covering of this casket with bone pieces in a specific way. Pieces cannot overlap and intersect the border of the covering. Archeologists are afraid to break the casket, so they just want to try all possible ways of tiling. Your task is to calculate the number of such ways.

Input

The only line of the input contains two space-separated integers W and H— the length and the width of the casket's covering ($1 \leqslant W, H \leqslant 100$).

Output

Output the number of ways of tiling modulo $10^9 + 7$.

Example

aztec.in	aztec.out
3 4	11

Problem C. Algorithm Complexity

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

Petr wants to use his own algorithm to solve a very important problem for a directed graph G with n vertices and m arcs. Unfortunately, Petr cannot calculate a complexity of his algorithm. He only knows that the complexity depends on the order of growth of value $F(N)$ which denotes the number of walks of length N from vertex s to vertex t in G. Petr wants to bound $F(N)$ with a polynomial of minimal degree, that is, to find the minimal non-negative integer k such that for some fixed number C inequality $F(N) \leqslant CN^k$ holds for any positive N. Help him to do it.

Input

The first line contains 4 space-separated integers n, m, s, t ($1 \leqslant n, m \leqslant 100\,000$). The vertices are numbered 1 to n. Each of the next m lines contains two space-separated integers — numbers of starting and ending vertices of the current arc. The graph doesn't contain multiple arcs but may contain loops.

Output

Output the minimal integer k which satisfies the problem statement. If there are no such numbers, output «-1».

Examples

complexity.in	complexity.out
2 3 1 2 1 1 1 2 2 2	1
3 6 1 2 1 2 2 1 1 3 3 1 2 3 3 2	-1

Problem D. Flags for Provinces

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

The government of Cuckooland decided that each province of such a huge country should have its own flag. A famous painter Cuckooshkin was told to create all flags. It is known that Cuckooshkin uses only N different colors in his paintings. According to the government's plan, every two flags of provinces should have at least one common color, which will symbolize integrity of the country. On the other hand, the painter wants to make these flags as varicoloured as possible, so he doesn't want any colour to occur in three or more flags. What is the maximal number of flags Cuckooshkin can create without breaking neither his own nor government's requirements?

Input

The first line contains the only integer N ($3 \leqslant N \leqslant 1000$) — the number of colors the painter can use. The colors are numbered 1 to N.

Output

In the first line output the integer K — the maximal number of flags the painter can create. Each of the next K lines should contain description of the next flag: first, the number of colors used in it and then the numbers of these colors. All integers in this line should be separated by single space.

Example

flags.in	flags.out
4	3 2 1 2 2 1 3 2 2 3

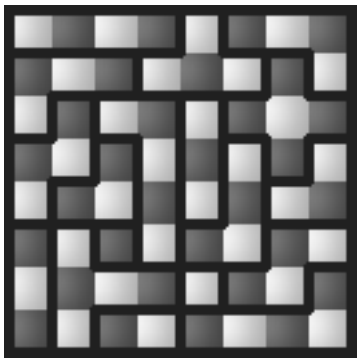
Problem E. Loyd's Problem

Input file:	loyd.in
Output file:	loyd.out
Time limit:	7 seconds
Memory limit:	256 megabytes

Sam Loyd is a famous american puzzle creator. One of his most famous puzzles is «The 15 puzzle». Also he is an author of many chess puzzles and cutting problems. Today you can try to solve his problem about cutting a chessboard.

You are given an $n \times n$ chessboard. Your goal is to cut it into the maximal number of different pieces. Each piece must consist of one or more cells and represent a side-connected region. If one piece can be obtained from another by a sequence of rotations then these pieces are considered equal. For example, there are two one-cell pieces: black cell and white cell and only one two-cell piece.

Here is one of possible solutions of the original Loyd's problem about cutting an 8×8 chessboard in 18 different pieces:



Input

The first line contains the only integer n — the length of the chessboard side ($1 \leqslant n \leqslant 30$).

Output

In the first line output the maximal number of different pieces the chessboard can be cut into. Then output the required cutting: n lines with n lowercase latin letters in each of them. Each piece must consist of the same letters, one letter can be used for representing several pieces, but every two pieces sharing a common side must be represented by different letters (see example for further clarification). If there are several optimal cuttings, you can output any of them.

Example

loyd.in	loyd.out
8	18 aaaacaaa bbbcccba baddabbb aacdacba accdacaa babdccbb baaadbba babbaaaa

Problem F. Killer Repetitions

Input file:	`repetitions.in`
Output file:	`repetitions.out`
Time limit:	5 seconds
Memory limit:	256 megabytes

Federal Security Agency is extremely interested in the loyalty of its special agents. To provide this loyalty they developed a «killer words» technology: if an agent doesn't execute orders anymore, it is enough to pronounce a special word to activate a bomb in the brain of the agent and eliminate him.

The bomb should not be activated accidentally, so the word should be quite special: it should contain only first M letters of english alphabet and should be a K-*repetition*, that is, it should be possible to represent it as a concatenation of K equal words. Moreover, to exclude the possibility of killing unnessecary agents, no proper substring of a killer repetition can be a killer repetition itself. Your task is to calculate the number of killer repetitions which consist of at most N letters.

Input

The first line contains space-separated integers M, K, N ($1 \leqslant M \leqslant 18$, $2 \leqslant K \leqslant 5$, $1 \leqslant N \leqslant 22$).

Output

Output the required number of killer repetitions.

Example

repetitions.in	repetitions.out
3 2 4	9

Problem G. Work for Robots

Input file:	`robots.in`
Output file:	`robots.out`
Time limit:	2 seconds (4 seconds for Java)
Memory limit:	256 megabytes

There are N robots on planet PTZZZ. Some of the robots are friends, and some of them are not.

Once a day some of the robots go to work and all the other robots go to a theme park and have fun. At least one robot should go to work. An administrator-robot decides who should go to work and who should have fun. The work is so important for robots that the first day when the administrator-robot made his decision was named the First day of the World.

If it turns out that the group of robots that goes to work is the same as the group in any day before, the administrator-robot will rust of sadness. Moreover, the law doesn't allow the administrator-robot to form a working group in such a way that there will be a pair of robots in this group that are not friends.

The administrator-robot doesn't want to rust, so since the first day he tries to form a different working group. However, the administrator-robot will rust sooner or later. Your task is to calculate the day number when this will happen.

Input

The first line contains an integer N — the number of robots on PTZZZ ($1 \leqslant N \leqslant 50$). Each of the next N lines contains N digits. j-th digit in i-th line is «`1`» if i-th and j-th robots are friends, and «`0`» otherwise. It is guaranteed that i-th digit in i-th line is equal to zero, and j-th digit in i-th line is equal to i-th digit in j-th line.

Output

Output the day number the administrator-robot will rust in.

Example

robots.in	robots.out
6 011100 101100 110100 111000 000001 000010	19

Problem H. Salary for Robots

Input file:	`salary.in`
Output file:	`salary.out`
Time limit:	5 seconds
Memory limit:	256 megabytes

There are N robots on planet PTZZZ. Each robot has its own unique rank — an integer from 1 to N, and should execute all orders from robots with a higher rank.

Once a month all robots get their salary: a positive integer number of credits, not exceeding K. The salary is paid by an accountant-robot. Salary is so important for robots that the first month when all the robots got their salary was named the First month of the First year. There are P months in the year on PTZZZ, so the robots get their salary P times a year.

The salary paid to each robot can be different in different months. If it turns out that all the robots get exactly the same salary as in any month earlier, the accountant-robot will rust of sadness. What is more, the law doesn't allow the accountant-robot to pay salary in such a way that there will be a triple of robots (A, B, C) with rank of A more than rank of B, rank of B more than rank of C and the salary of A less than the salary of B and the salary of B less than the salary of C.

The accountant-robot doesn't want to rust, so since the First month of the First year he tries to pay salary in different ways. However, the accountant-robot will rust sooner or later. Your task is to calculate the month number when this will happen.

Input

The only input line contains three space-separated integers N, K and P — the number of robots on PTZZZ, the maximal possible salary and the number of months in a year, respectively ($1 \leqslant N \leqslant 1000$, $1 \leqslant K \leqslant 200$, $2 \leqslant P \leqslant 10^9$).

Output

Output the month number the accountant-robot will rust in. Months are numerated 1 to P.

Example

salary.in	salary.out
3 3 20	7

Problem I. Sniper Shot

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

There is a sniper at point S. His mission is to eliminate an enemy of the state, who rides his bicycle along a straight line from point A to point B. The bullet flies along a striaght line with infinite speed. There are N rectangular parallelepiped-shaped skyscrapers in the city. The bullet can't fly through the skyscraper but can touch its border. Of course, the sniper will make a deadly shot as soon as possible. Your task is to calculate the coordinates of the enemy at the moment of the shot.

Input

The first line contains space-separated coordinates of S: sx, sy, sz ($sz \geqslant 0$). The second line contains space-separated coordinates of points A and B: ax, ay, bx, by. The enemy of the state moves on the surface of earth, so his z-coordinate is always equal to zero. The third line containts an integer N ($0 \leqslant N \leqslant 100$). Each of the following N lines contains space-separated numbers lx, ly, rx, ry, h ($lx < rx$, $ly < ry$, $h > 0$) — coordinates of the opposite corners of the bottom of the current skyscraper and its height. The sides of the skyscrapers are parallel to the corrdinate axes. All coordinates and heights are integers and don't exceed 100 by their absolute values. It is guaranteed that no two skyscrapers have common points, the point S doesn't lie inside or on the border of the skyscraper and the segment AB doesn't have common points with any of the skyscrapers.

Output

If the enemy of the state cannot be eliminated, output «`Impossible`». In the other case output the coordinates of the enemy of the state precise up to 10^{-7}.

Examples

sniper.in	sniper.out
0 0 2 -4 4 4 4 2 -3 2 -1 3 10 1 -1 4 2 20	-1.3333333333 4.0000000000
0 0 2 4 1 4 -1 1 1 -1 3 1 10	Impossible

Problem J. Square Country 5

Input file:	square5.in
Output file:	square5.out
Time limit:	3 seconds
Memory limit:	256 megabytes

The first arithmetical operation taught to the children of the Square country is the calculation of squares of positive integers. At the first lesson the children are provided with «easy» numbers, calculating a square of which can be done by writing a few digits in front of them (i.e. 76 is an easy number because $76^2 = 5776$). The task shouldn't be too difficult, so the easy numbers shouldn't contain more than N digits. How many different easy numbers can teachers prepare for the first lesson?

Input

The only input line contains an integer N ($1 \leqslant N \leqslant 2000$) — the maximal length of the easy number the children can be provided with.

Output

Output a single integer — the number of different easy numbers consisting of at most N digits.

Example

square5.in	square5.out
1	3

Problem K. Turning Turtles

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

Military built a rectangular training ground of $W \times H$ cells to train battle turtles. Some of the cells are passable for turtles, and some of them are not. Turtles can move only parallel to the sides of the training ground. The ground is constructed in such a way that there is exactly one way to get from one passable cell to another passable cell without visiting any cell twice. It is known that turtles can run in a straight line very fast, but it is difficult for them to turn 90 degrees. So the complexity of the route is caluclated as the number of turns the turtle will make on its way from the initial to the final cell of the route. Your task is to write a program which will calculate the complexity of the route knowing its initial and final cell.

Input

The first line contains two space-separated integers H and W — the lengths of the ground sides ($1 \leqslant W \times H \leqslant 100\,000$). Then follows the map of the polygon — H lines with W symbols in each. Symbol «#» stays for a passable cell and «.» stays for a non-passable cell. Line number $H + 2$ contains an integer Q — the number of routes you have to calculate the complexity for ($1 \leqslant Q \leqslant 50\,000$). Each of the next Q lines contains four space-separated integers: the number of row and the number of column of the initial cell of the route, the number of row and the number of column of the final cell of the route, respectively. It is guaranteed that the initial and the final cells of the route are passable. Rows are numerated 1 to H from top to bottom, columns are numerated 1 to W from left to right.

Output

For each route output its complexity in a separate line.

Example

turtles.in	turtles.out
5 4 .#.. ###. ..## .##. 4 1 2 2 1 2 3 4 3 4 2 3 4 1 2 4 2	1 0 2 3