UVA [redacted] Association
[redacted] for Computing Machinery
[redacted] 13^{th} Annual [redacted]
High School Programming Contest

Problem A
No More Four!
Time limit: 13 seconds

The Hiding Spies Public Committee (HSPC) just found a vulnerability in their computers! The details of the vulnerability are still classified, but the HSPC has told you that it is related with computing on the number 4. The evil Association for Cruel Machinations (ACM) has found out about the vulnerability and has begun sending thousands of messages each with a single digit 1-9. You are tasked with writing a program that will filter through all these messages and identify which ones you should block and which ones are safe.

Input

The only line of the input contains a single integer x $(1 \leq x \leq 9)$ - the message sent by ACM.

Output

Output one line of output containing a single string. If the input is 4, then output “block”. Otherwise, output “safe”.

Sample Input 1	Sample Output 1
`1`	`safe`

Sample Input 2	Sample Output 2
`4`	`block`

Problem B
Cars
Time limit: 2 seconds

To ensure the safety of the public, the Hiding Spies Public Committee (HSPC) has a unique system for sending classified messages to agents across the globe.

First, a message is broken up into n fragments numbered from 1 to n. These pieces will be delivered separately by n agents. Each agent receives a fragment of the message in their car. The agents will drive together in a line to deliver their portion of the message to the receiving agent. While they are driving, agents may pass each other. Because of this, once the fragments are delivered, they may be out of order.

In order to recover the original message, HSPC provides receiving agents with two important pieces of information. First, HSPC sends the initial ordering of the cars when they receive their fragments of the message. Second, HSPC sends sensor information from each of the agents' cars. All HSPC cars are equipped with a sensor that will detect when that agent passes the agent in front of them while driving. Due to a lack of funding, these sensors sometimes will malfunction. In particular, the car in the front of the line may sometimes fire off even though there is no car in front of them to pass. Agents are told to ignore these incorrect signals.

Unfortunately, Agent B just got started on the job and is having trouble figuring out the final order of the messages! Can you help them by devising a program that deciphers the final message ordering for Agent B to use in future missions?

Input

The first line of the input contains n, q, $(1 \leq n, q \leq 1\,000)$ - the number of message fragments and the number of signals from the car sensors **including malfunctions**.

Then, one line follows, the initial ordering of the messages in the n cars. The first number in the array is the message carried by the car at the front of the line.

The next q lines contain an integer a_i, $(1 \leq a_i \leq n)$. This indicates that car carrying message a_i passes the car that is currently in front of it. All the signals are given in the order that they occurred including the malfunctions.

Output

Print one line, the final ordering of the fragments.

Sample Input 1

```
5 2
1 2 3 4 5
2
3
```

Sample Output 1

```
2 3 1 4 5
```

Sample Input 2

```
5 2
5 1 3 2 4
2
3
```

Sample Output 2

```
5 1 3 2 4
```

Sample Input 3	Sample Output 3
2 3 2 1 2 2 2	2 1

Problem C
TopSecret
Time limit: 1 second

Due to some unfortunate budget cuts, the agency has had to revisit its classification scheme. Starting immediately, there is only sufficient funding to secure a single "TopSecret." Sadly, this means that some previous secrets will need to be declassified. To assist in this, the agency would like some automated means of determining how close some secret is to being "TopSecret."

The agency's current scheme for storing the classification level of a secret does not care about order: A secret which is "SectrepoT" is considered to be the same level of secrecy as a secret which is "TopSecret." A secret which is "MidSecret" is not subject to anywhere close to the same level of secrecy as a "TopSecret." Some secrets are *close* to being a "TopSecret"; that is, they differ from "TopSecret" in the cases of some letters. While a secret which is "topSecreT" is the same as a secret which is "TopSecret," a secret which is "topSecret" is considered to be distance 1 from a "TopSecret." Likewise a secret which is "TOPSECRET" is distance 7 from a secret which is "TopSecret."

Given a string, determine how far it is from being "TopSecret." If it is not "TopSecret" after any amount of rearrangements and case changes, print "No".

Input

The only line of the input contains a string of length at most 10 and consists only of lower and uppercase letters of the English alphabet.

Output

Output the minimum number of case changes to make the string equal to some permutation of the string "TopSecret" (without the quotes), or print "No" (also without the quotes) if it is not possible.

Sample Input 1	Sample Output 1
`TopSecret`	`0`

Sample Input 2	Sample Output 2
`SectrepoT`	`0`

Sample Input 3	Sample Output 3
`MidSecret`	`No`

Sample Input 4	Sample Output 4
`topSecreT`	`0`

Sample Input 5	Sample Output 5
`topSecret`	`1`

Sample Input 6	Sample Output 6
`TOPSECRET`	`7`

Problem D
Fight Scene
Time limit: 2 seconds

For your next mission, your job is to create gossip at an important political dinner hosted by the Association for Cruel Machinations (ACM) executive Dr. P in his malevolent mansion. While you are only supposed to spread information on the evildoing of ACM, you do not want to cause havoc potentially injuring prime political candidates that support the HSPC. From your previous mission, you were able to secure a guise from one of his cronies and enter the building with ease. You begin making benevolent accusations of dinner members, leading to some discomfort and visible anger between groups.

Your plan is working smoothly, perhaps too smoothly... when a fight roars at the dinner table! With everyone screaming and shouting at each other, you keep your calm and analyze the strengths of everyone. You see that there are n guests at the dinner table. Those who are weaker at the dinner party will group together to overpower stronger guests. You want to find out who will need help protecting themselves and who will be able to defend against all the guests weaker than them.

Given that there are n guests each with some strength a_i, determine which of the dinner guests will not be able to fend off the combined strength of all other guests with strength weaker than them.

Input

The first line contains one integer n, $(1 \leq n \leq 100\,000)$ - The number of dinner guests.

The following line contains of n integers a_i, $(1 \leq a_i \leq 10^9)$ - The strengths of the n guests. It is guaranteed that all elements are distinct.

Output

Print a string containing n characters. The i-th character represents if the strengths of all weaker guests sum to a value equal to (E), greater (G), or less than (L) guest i.

Sample Input 1	Sample Output 1
5 1 2 3 4 5	GGELL

Sample Input 2	Sample Output 2
5 3 1 4 5 2	EGLLG

Sample Input 3	Sample Output 3
4 1000000000 1234 10 1	GGGG

Problem E
Dodge
Time limit: 5 seconds

Your next mission is to take down the nefarious Agent E, who will be traveling on the Waterfall Causeway to deliver critical information that will hinder the Hiding Spies Public Committee (HSPC). You arrive at the midpoint of the bridge where you await for a shaded car to approach. As the sun sets, you spot the designated vehicle and begin mowing it down! You are met with retaliation as the car speeds down the bridge. You jump into your car and begin pursuing Agent E to ensure that they don't get away! Unfortunately, you are still met with a barrage of gunshots as his accomplices try to stop you from following them. How will you be able to avoid all these gunshots?

Luckily, the HSPC has a special gadget in all their cars that will display the optimal way for the car to move in order to dodge incoming fire. The car will display a string of n characters where each character is either "L" or "R". The i^{th} character of the string indicates that you must move to the left if the character is an "L" or to the right if the character is an "R". Unfortunately, the HSPC R&D team is not funded well enough, so the computer is not able to tell you how far to the left or right you should dodge.

You are currently on a bridge that is k units wide. You must determine some valid integer starting position s and a sequence of n **strictly positive** integers that represent the distances that you will move for each dodge in the dodging sequence. To keep up with Agent E, you must stay on the bridge. Determine any possible starting position and list of distances such that you are able to stay on the bridge and perform the sequence of dodges you are given or determine that it is impossible and you should abandon the mission.

Input

The first line contains two integers n and k, $(1 \leq k, n \leq 100\,000)$ - The length of the dodging sequence and width of the bridge.

The following line contains a string of length n only containing the characters "L" and "R" representing the moves of the dodging sequence in order.

Output

The first line of output should contain the string "possible" if it is possible to perform the dodging sequence within the width of the bridge or the string "impossible" if it is impossible.

If it is possible to perform the dodging sequence, output two additional lines that represent any possible way to perform the dodging sequence. In the first line, output the starting position s, $1 \leq s \leq k$. On the second line, output n positive integers $a_1, a_2, \ldots, a_n$, where a_i is the number of lanes left or right moved during the i^{th} dodge.

Sample Input 1	Sample Output 1
11 5 RRRLRRRLRRR	possible 1 1 1 1 3 1 1 1 3 1 1 1

Sample Input 2	Sample Output 2
5 3 RRRRR	impossible

Problem F
Passwordle
Time limit: 5 seconds

Your job is to retrieve the details of a heist that the Association for Cruel Machinations (ACM) has planned to execute from their main headquarters. With your extremely excellent spy skills, you pass through the lasers like nothing and slide by the guards without a trace! As you arrive to the vault, the only thing that is stopping you from retrieving the goods is a digital lock that will signal an alarm once you fail more than 6 attempts.

Fortunately, the Hiding Spies Public Committee (HSPC) has given you a lock cracker that will be able to break the lock with ease! However, due to some budget constraints in the HSPC, you realize that they gave you the v1 prototype which can only crack the password once you are able to determine at least 20% of the letters. You know that the ACM only uses uppercase latin letters for their passwords, so you will make up to 6 guesses of n letters. After each guess you make, the lock cracker can determine which characters in the password you chose correctly and which were incorrect. Will you be able to break the lock without the alarm going off?

Interaction

Initially, you will receive a single line of input containing an integer n ($10\,000 \leq n \leq 100\,000$), the length of the password you are guessing. The sample cases are special and have $n < 10\,000$ so it is easy to understand interaction. Your code will not be judged against sample cases and all hidden cases will satisfy ($10\,000 \leq n \leq 100\,000$).

Your program should output guesses for the correct password in the form of a string of uppercase Latin letters. After making a guess, make sure to flush standard out.

After each guess, you will receive a string of length n from standard in. Each character of the string will be either 'X' representing a correct guess for that letter or '_' representing an incorrect guess for that letter. If your guess correctly identifies 20% or more of the password, then your program should exit and you will receive an ACCEPTED verdict. If you use more than 6 guesses, or if your program guesses after you have already correctly identified 20% or more of the password, your program will be terminated and you will receive a WRONG ANSWER verdict.

The password string is fixed at the start of any case and will remain the same during all queries.

Sample Interaction 1

Read	Write
6	
	AAAAAA
______	
	BBBBBB
____X_	
	CCCCCC
_X____	
	ACAABA
_X__X_	

Sample Interaction 2

Read	Write
5	
	ABCDE

```
XXXX_
```

Problem G
Designating Depots
Time limit: 2 seconds

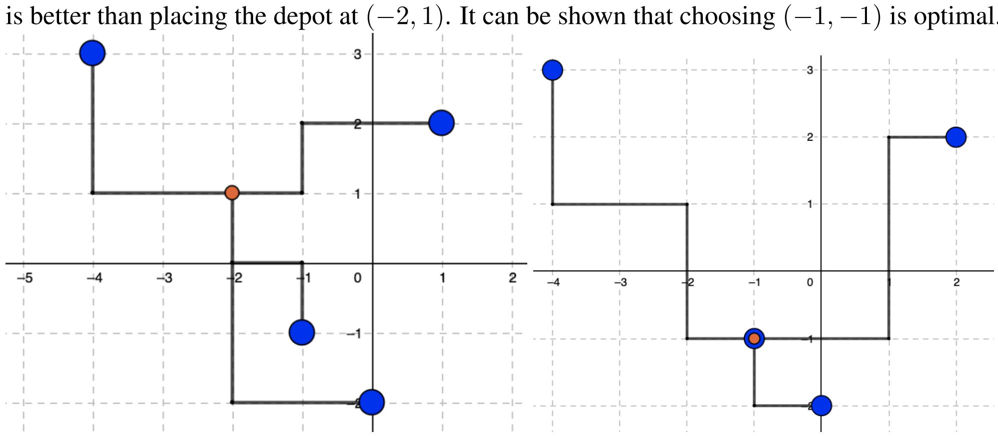
Recently, the Hiding Spies Public Committee (HSPC) has been having issues with distributing supplies. There are n HSPC bases located on integer coordinates of the two-dimensional coordinate plane. The HSPC supplies all of their bases from a central depot that also has integer coordinates. Supplies are sent to bases from the depot through a railway system where there are vertical and horizontal railways at every integer x and y value. The cost of transporting from the depot to a base is the minimum distance that a train must travel to go from the depot to the HSPC base. See the diagrams for the first sample for clarification.

In the current layout of bases, the headquarters is very far from the depot while many of the less important hideouts are close to the depot. Clearly this is very inefficient, so you have decided to restructure the HSPC bases. In the most efficient layout of bases, more important bases will be closer to the depot and less important bases will be further from the depot.

You are given the n locations of the HSPC bases on the two-dimensional coordinate plane. It would be far too expensive to rebuild a base at a new location, but it is easy to move the depot location to any point **including one that is also occupied by a base**. Additionally, all HSPC bases are designed to be architecturally identical, so after you choose a depot location, the closest base to the depot will become the headquarters, the second closest base will be the second most important base, etc. Because the headquarters is the most important base, over all placements of the depot, you want to choose the point that minimizes the cost for the headquarters. If there are multiple points that minimize this cost, then you want to choose the one that also minimizes the cost for the second most important base. You want to continue minimizing like this over all bases. Can you find the optimal depot location?

Sample Explanation

We recommend viewing the diagrams on the electronic version of the problems, so that color differences are clear. Shown below are two possible placements for the depot (small orange dot). The large blue dots represent HSPC base locations and black lines represent an example shortest path for a train traveling from the depot to each base. In the first placement, the depot is placed at $(-2, 1)$. The closest base is $(-1, -1)$, so it becomes the headquarters with a distance of 3 from the depot. The costs for the other 3 bases are 4, 4, and 5. In the second placement, the depot is placed at $(-1, -1)$. Note that the depot **can be placed on a base**. The distances of the four bases are 0, 2, 6, and 7. As we can see, the headquarters is a distance of 0 from the depot with this placement, so placing the depot at $(-1, -1)$ is better than placing the depot at $(-2, 1)$. It can be shown that choosing $(-1, -1)$ is optimal.



Input

The first line contains a single integer n $(n \leq 1\,000)$ - The number of sites.

Then follow n lines representing the locations of the sites. The i^{th} line contains two integers x_i and y_i $(|x_i|, |y_i| \leq 10^9)$ which are the x and y coordinates of the i^{th} site.

Output

Print a single line containing two integers x and y, which are the coordinates of the optimal depot location. If there are multiple optimal depot locations, then output any of them.

Sample Input 1

```
4
-4 3
1 2
0 -2
-1 -1
```

Sample Output 1

```
-1 -1
```

Sample Input 2

```
1
-1000000000 -1000000000
```

Sample Output 2

```
-1000000000 -1000000000
```

Problem H
Follow the Bouncing Laser
Time limit: 5 seconds

For your next mission, you have been tasked with eliminating Association for Cruel Machinations (ACM) Agent N that is planning to rob the Notoriously Astounding Crown (NAC) artifact in the center of the Mirror Museum. The Hiding Spies Public Committee (HSPC) has given you a laser gun capable of eliminating the target. The gun shoots out a laser that is able to bounce off of the mirror walls and scorch your enemies by surprise. However, the laser is at its strongest after bouncing off mirrors exactly k times. You will also have to be careful and not shoot any artifacts that are in the museum. Even slightly grazing an artifact with the laser may cause it to catch on fire, so you must be careful.

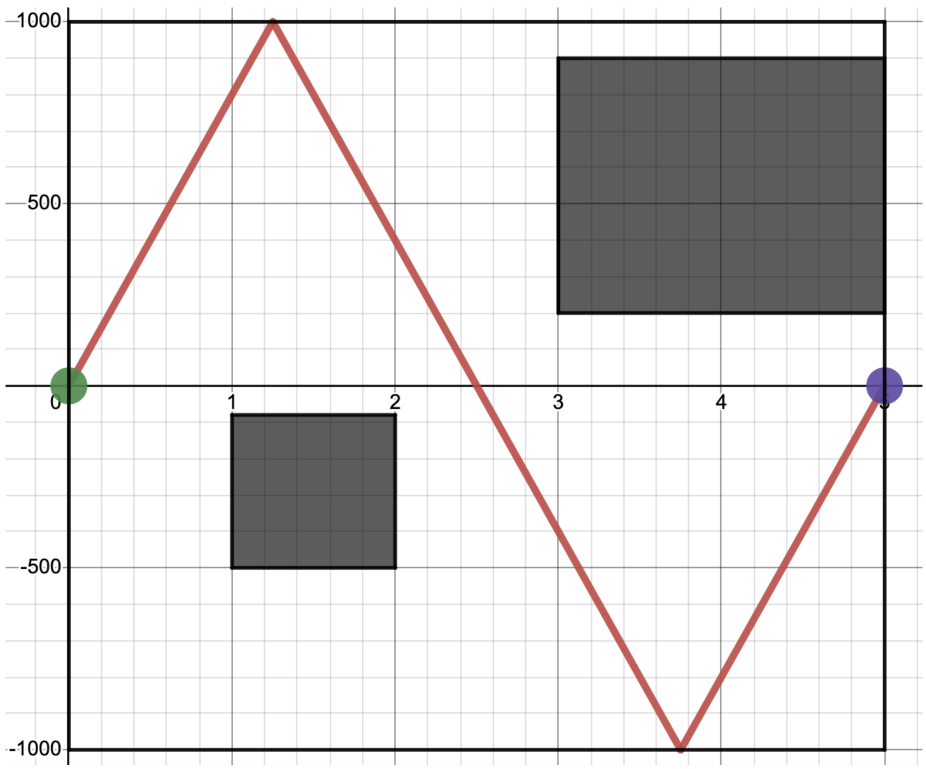
You enter the museum at night awaiting for the agent to strike at any moment. While standing in the center of the museum, you catch a glimpse of the perpetrator bringing themself down by rope to rob the NAC!

The room you are in is a rectangle of length t and height 2000. Along the top wall is a horizontal mirror at $y = 1000$ and the bottom wall has a horizontal mirror at $y = -1000$. The left and right walls **are not mirrors**. You are in the middle of the left wall of the room at the point $(0, 0)$ and Agent N is on the right wall at $(t, 0)$. In addition you are given n artifacts, which are axis-aligned rectangles that are completely contained within the room.

Determine if it is possible to shoot Agent N with your laser gun after exactly k bounces without hitting an artifact and determine what direction you must shoot if it is possible. The laser is considered to hit an artifact if the path of the laser intersects any point **including corners** on the border of the artifact.

Sample Explanation

A diagram of the first sample is shown below. You are located and point $(0, 0)$ and Agent N is at $(5, 0)$. Firing the laser with a slope of $800/1$ will avoid the artifacts (boxes) and will hit Agent N after exactly 2 bounces.



Input

The first line of input contains three positive integers t, n, and k $(1 \leq t, n, k \leq 1\,000)$ - The length of the room, the number of artifacts, and the number of bounces the laser must make, respectively.

n lines follow. The i^{th} of these lines will contain four integers x_i y_i w_i h_i representing the i^{th} artifact. x_i and y_i are the coordinates of the upper-left corner of the i^{th} artifact. w_i and h_i are the width and height of the i^{th} artifact.

Output

On the first line of output, print the string "possible" if it is possible to shoot agent N or print "impossible" if it is impossible.

If it is possible to shoot agent N, then print a second line containing a fraction in reduced form that is the slope in which you would shoot the laser in order to hit agent N after exactly k bounces.

Problem I
Ciphers
Time limit: 2 seconds

Recall the definition of a *monoalphabetic substitution cipher*: a proper monoalphabetic substitution cipher is a 26×2 table with entries all uppercase Latin letters such that no letter occurs more than once in a given column. Two examples are given below:

A	N
B	O
C	P
D	Q
E	R
F	S
G	T
H	U
I	V
J	W
K	X
L	Y
M	Z
N	A
O	B
P	C
Q	D
R	E
S	F
T	G
U	H
V	I
W	J
X	K
Y	L
Z	M

Sample 1: ROT13 Cipher

A	B
B	C
C	A
D	E
E	F
F	D
G	H
H	I
I	G
J	K
K	L
L	J
M	N
N	O
O	M
P	Q
Q	R
R	P
S	T
T	U
U	S
V	W
W	X
X	V
Y	Z
Z	Y

Sample 2: Another Cipher

Note that a monoalphabetic substitution cipher need not necessarily have the first column in order, that is, there are up to $26! = 403\,291\,461\,126\,605\,635\,584\,000\,000$ valid configurations of the first column. A monoalphabetic substitution cipher is usually read left to right, for example, to encode the letter 'X', one would find 'X' in the first column (in both of the samples, this happens in the 24^{th} row) and write down the other letter in the same row. So, 'X' in ROT13 is 'K', while 'X' in the second cipher is 'V'. To encode a string, one just encodes each letter one-by-one; for example, "HSPC" in ROT13 is "UFCP".

Your informant, Mr. Jock of the High Spies Public Committee (HSPC), has come into possession of an enemy cipher from the Association for Cruel Machinations (ACM). However, the enemy counterintelligence caught wind of his plan and hacked the program he used to send you the cipher by silently flipping some of the rows in the cipher hoping you would not notice. Thus, some of the rows in the table may end up getting flipped – more precisely, each row is flipped independently at random with probability $\frac{1}{10}$. What are the odds that, after flipping, the table still forms a valid monoalphabetic substitution cipher?

Input

The first line contains one integer t ($1 \leq t \leq 100$), the number of ciphers.
Then, t lines follow. On the i^{th} line is a single string: the message "MRJOCKTVQUIZPHDBAGSFEWLYNX" encoded in the i^{th} cipher.

Output

Print t lines. On the i^{th} line, print the **exact** probability that, after the random flipping, the table of the i^{th} cipher is still a valid monoalphabetic substitution cipher. You must truncate additional trailing zeroes (0.0 and 1.0 will still be accepted). If the answer is not 1, you must include exactly one zero before the decimal.

Sample Input 1

```
3
ZEWBPXGIDHVMCUQONTFSRJYLAK
NPKMALUWRSGYQIECBHTDFXJZOV
MRJOCKTVQUIZPHDBAGSFEWLYNX
```

Sample Output 1

```
0.07578444614164591651397632
0.066129727535314642
1.0
```

Problem J
Dictionary Attack
Time limit: 5 seconds

You have intercepted a message from the Association for Cruel Machinations (ACM)! However, some of the letters have been lost. Fortunately, agents at HSPC have found the dictionary of all the words that ACM uses to craft their messages, so you may be able to recover its contents.

You have a length n secret message with some characters missing. We also have a list of m known words each of length up to length 10. We want to map the words onto the secret message to recover the missing letters.

Input

The first line of input contains a single integer n ($1 \leq n \leq 100\,000$) - the length of the message. The second line contains the string of length n containing lowercase Latin letters representing the message.

The third line contains a single integer m ($1 \leq m \leq 1\,000$) - the number of words in the HSPC dictionary. m lines follow. Each line contains a word in the HSPC dictionary. Each word is a string of lowercase Latin letters of length at most 10.

Output

In the first line of output, print the string "crackable" if there exists a message consisting only of words from the ACM dictionary. Otherwise print "uncrackable".

If the message is crackable, then print any possible message that only contains words from the ACM dictionary.

Sample Input 1

```
9
t?ps??re?
4
top
bottom
secret
message
```

Sample Output 1

```
crackable
topsecret
```

Sample Input 2

```
9
t?ps??re?
3
bottom
secret
message
```

Sample Output 2

```
uncrackable
```

Sample Input 3	Sample Output 3
9 notsecret 2 top secret	uncrackable

Problem K
Block Coding
Time limit: 13 seconds

The Hiding Spies Public Committee (HSPC) is testing a new system for encoding secret messages. Essentially, the message will be encoded as a sequence of floating “3-D Blocks”: each letter will be written in three dimensions by breaking it down into axis-aligned non-intersecting (but possibly touching) rectangular prisms. The corners of these prisms will all be at integer points.

Due to budget constraints, some of the HSPC codebooks differ in their representations of letters: one may represent the same region (a 3-D L) by a $1 \times 1 \times 3$ prism with a $1 \times 1 \times 1$ cube adjoined to the right at the base, or by a $1 \times 1 \times 2$ prism sitting atop a $1 \times 2 \times 1$ prism (see the samples for a better visualization of this).

Two representations are considered the same when each point in the entire 3-D space is contained in some prism in the first representation if and only if it is contained in some prism in the second.

Input

The input will contain exactly two block codings. Each block coding consists first of a single line with the number of prisms n ($0 \le n \le 100\,000$) in that coding. Then, on each of the following n lines is a description of a prism as six integers $-10^9 \le x, y, z, x', y', z' \le 10^9$ where (x, y, z) and (x', y', z') are any two opposite corners on that prism. All prisms are guaranteed to be nondegenerate; that is, $x \ne x'$ and $y \ne y'$ and $z \ne z'$.

No two prisms in the same block coding have a nondegenerate intersection. The number of blocks in each code may differ.

Output

Output on one line “same” if the two codings represent the same region in 3-D space and “different” otherwise.

Sample Input 1

```
2
0 0 0 2 1 1
0 0 1 1 1 3
2
0 1 3 1 0 0
1 0 0 2 1 1
```

Sample Output 1

```
same
```

Sample Input 2

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

Sample Output 2

```
different
```

Problem L
Self Destruct
Time limit: 5 seconds

Operatives and connections have been getting compromised like mad! The Hiding Spies Public Committee (HSPC) is fairly certain that at this point, even the most secure field networks will be brought down within days of bringing them up. The HSPC is wondering how to foresee these risks while still transferring the most essential information for as long as possible.

For security reasons, the HSPC tries to minimize the total number of contacts that their operatives have, so that when operatives are compromised, the process spreads slowly. That is, if there are n spies in the field and each spy i knows $d(i)$ people, the sum of $d(i)$ over all i is exactly $2n - 2$. The notion of "knowing" is symmetric; that is, if a knows b, then b also knows a. Spies can transfer almost any information to anyone they know instantly, and that information can then be spread; initially, each spy will be able to instantly transmit information to any other spy by sending some communication of the form "psst, pass this on to a: 'psst, pass this on to b: ...'"

However, when a spy suspects they are about to be compromised, the process is slightly different. A spy who initiates self-destruct instead sends the following message to **everyone** they know:

> Well, [sappy message here]. [sic]

The spy then responsibly handles the situation through means not disclosed to us; however, we can reasonably conclude that working with them, **including communicating through them**, is no longer possible. After taking exactly 1 second to absorb this message, all of that spy's contacts then initiate the same self-destruct process.

Of course, after some spy initiates this self-destruct process, the whole network will eventually be destroyed. However, some pairs of spies may still be able to communicate for some time before losing the ability to do so. The HSPC wants to know: given a pair of spies i, j, if a random spy self-destructs, what is the expected amount of time until i is no longer able to send information to j?

Input

The first line contains two space-separated integers: the number of spies n ($1 \leq n \leq 200\,000$) and the number of queries q ($1 \leq q \leq 200\,000$).

The next n lines describe the structure of the spy network: the first integer on the i^{th} line is $d(i)$ ($1 \leq d(i) \leq n - 1$), the number of other spies spy i knows. The remaining $d(i)$ integers on the i^{th} line are the list of distinct spies that spy i knows. Here spies are numbered 1 through n, and no spy is considered to "know" themself.

It is guaranteed that the sum of $d(i)$ over all i is exactly $2n - 2$.

The remaining q lines each contain a pair of spies a, b ($1 \leq a < b \leq n$) for which HSPC wants to know the answer to the above question.

Output

Output a single line for each query. For each query, calculate the expected amount of time before i is no longer able to send information to j. Because this value may not be an integer, print the answer multiplied by n.

Sample Input 1

```
6
2 2 3
2 1 4
3 1 5 6
1 2
1 3
1 3
3
1 2
4 6
5 5
```

Sample Output 1

```
6
1
12
```

Sample Input 2

```
5
1 2
2 1 3
2 2 4
2 3 5
1 4
2
1 5
3 1
```

Sample Output 2

```
0
3
```

Problem M
Directing Directors
Time limit: 5 seconds

Spy directors from across the Hiding Spies Public Committee (HSPC) each have important information to tell everyone across the agency! Due to the criticality of this information, it is imperative that the directors visit in-person to the hideouts to ensure no information gets leaked out to the public.

As the new intern for the HSPC, you have been assigned the critical task of figuring out how to schedule the director visits to each of the hideouts. Over the next n days, all n of the spy directors will each visit one of the n agent hideouts across the nation. An important HSPC policy is that no two directors may be in the same building at the same time. In the case of an infiltration or assassination, at most one director will be in danger.

You have been given an $n \times n$ matrix describing a partially filled schedule. The j^{th} element of the i^{th} row in the matrix contains the director who will visit the i^{th} base on the j^{th} day. If the value is a 0, then there is no director currently scheduled on that day, so you must fill that spot in. If a director is already scheduled in the matrix, they will be scheduled exactly n times. However, it is possible that previous interns have made mistakes and scheduled directors at two bases on the same day or the same base on two separate days. Your job is to figure out if the 0's in the schedule can be replaced with director numbers so that all n directors visit each base exactly once.

Input

The first line of input will contain a single integer n ($1 \le n \le 100$). n lines of n integers from 0 to n will follow. The j^{th} value of the i^{th} line contains the number of the director who is scheduled to visit the i^{th} hideout on the j^{th} day. If the value is 0, then there currently is no director scheduled for that hideout on that day. It is guaranteed that all directors will be either fully scheduled or not scheduled at all.

Output

In the first line of output, print "possible" if it is possible to schedule the directors so that every director visits every site exactly once. Otherwise, print "impossible."

If it is possible, print any valid scheduling with all 0s from the original schedule replaced with a value 1 to n.

Sample Input 1

```
10
0 3 0 1 10 8 0 0 6 0
3 8 6 0 0 10 0 0 0 1
0 1 3 0 8 0 6 10 0 0
8 0 1 6 3 0 10 0 0 0
0 0 0 10 0 6 0 3 1 8
0 6 0 0 1 3 0 8 10 0
1 0 10 0 0 0 8 0 3 6
10 0 0 8 0 1 3 6 0 0
6 0 0 3 0 0 0 1 8 10
0 10 8 0 6 0 1 0 0 3
```

Sample Output 1

```
possible
2 3 4 1 10 8 5 7 6 9
3 8 6 2 4 10 9 5 7 1
4 1 3 5 8 7 6 10 9 2
8 2 1 6 3 4 10 9 5 7
5 4 2 10 9 6 7 3 1 8
9 6 7 4 1 3 2 8 10 5
1 5 10 7 2 9 8 4 3 6
10 7 9 8 5 1 3 6 2 4
6 9 5 3 7 2 4 1 8 10
7 10 8 9 6 5 1 2 4 3
```

Sample Input 2	Sample Output 2
3 1 0 1 0 1 0 0 0 0	impossible

Problem N
Association for Censoring Matrices
Time limit: 13 seconds

Finally, you've made it to the center of the lair of the evil Association for Censoring Matrices (ACM)! At the heart of their headquarters is a prototype for an absolutely diabolical contraption: the matricioautocensorificator! Worse yet, you're now trapped! You'd like to covertly send a message requesting backup to the Hiding Spies Public Committee (HSPC), but the device is interfering with your ability to do so! The protocol for the HSPC SOS is *typically* as follows: each day while the HSPC has not arrived, you must send *any* 0-1 matrix of fixed dimensions $h \times w$. It may take multiple days for the HSPC to pinpoint your location. However, you are confident that it will take at most 17 days before the HSPC is able to pinpoint your exact location and send agents to your aid (hence you never have to send more than 17 matrices). Once they do so, you're sure they can help free you from the clutches of the fiendish matricioautocensorificator!

There's just one problem: the dastardly matricioautocensorificator prevents you from sending some parts of certain matrices again after sending that part once; that is, each day you send a transmission to the HSPC and they do not come to your aid, the devilish matricioautocensorificator's mind control module prevents you from ever again sending some nonoverlapping subset of the 3×3 subgrids of the transmission you most recently sent at the same indices in which you just sent them.

Send the matrices. Escape the clutches of the wicked matricioautocensorificator. The HSPC needs agents like you to survive their field missions, especially in light of the recent budget cuts.

Interaction

The judge will write to your program's standard input and read from your program's standard output. First, the judge will send the agreed-upon dimensions of the matrix h ($3 \leq h \leq 300$) and w ($3 \leq w \leq 300$). Then, the following will happen up to 17 times:

Your program will print a binary $h \times w$ matrix (without spaces) to standard output. Then your program will recieve a newline-separated list of pairs of indices r, c (0-indexed from the top left, $0 \leq r \leq h - 3$ and $0 \leq c \leq w - 3$), indicating that the 3×3 subgrid that your program printed with upper-left corner at row r and column c has been matricioautocensorificated. If at any point your program prints this 3×3 subgrid in the same location again, the matricioautocensorificator will kill you and you will thus receive a WRONG ANSWER verdict. All of the 3×3 subgrids in this list will not overlap, though they may overlap with previous or future subgrids.

The list of indices your program recieves will be length zero if and only if it is the last of the (up to 17) iterations.

Read	Sample Interaction 1	Write

Read:
```
3 7
```

Write:
```
0000000
0000000
0000000
```

Read:
```
0 0
-1 -1
```

Write:
```
0000000
0000000
0010000
```

```
0 0
0 4
-1 -1
```

```
0000000
0010000
0000001
```

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

```
0000000
0011000
0000001
```

```
-1 -1
```