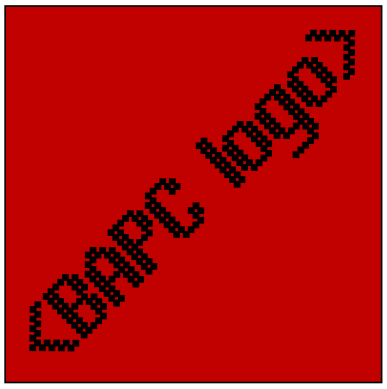
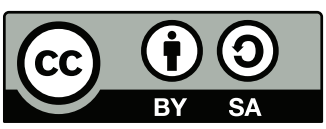
GCPC 2026

German Collegiate Programming Contest 2026



Problems

A	Attracting Attendees
B	Bye Bye Bilbo
C	Crosses and Circles
D	Delphi Danger
E	Egocentric Expedition
F	Fighting Fraud
G	Garbled Garden
H	Historical Hits
I	Incremented Itinerary
J	Junior Joining
K	Keeping Cows
L	Lyrical Leisure
M	Mirror Magic



A Attracting Attendees

Time limit: 4s

Lineup of the skybird fest.

You are the organizer of the first *Global Chart Pop Concert (GCPC)*, and you want to find a good lineup for your festival to become a hit.

You have already gathered a list of n bands that are available to perform at GCPC and a list of m people that consider going to the festival. From a quick market analysis, you know that a person will only attend the festival if at least half of their favourite bands perform. And even if they attend the festival they still watch the performances of only their favourite bands at the festival. Luckily, you have also gathered the favourite bands of each of the m people during your analysis. With this information, picking a good lineup might seem trivial, but there is one last catch. A band refuses to perform if less than c fans are watching since nobody likes a boring crowd.

To avoid the GCPC becoming a *flop*, you need to ensure that the crowd for each performing band is large enough. Obviously, a festival without any band is a complete flop.

Input

The input consists of:

- One line with three integers n, m, and c ($1 \leq n, m, c \leq 2 \cdot 10^5$), the number of bands, the number of people, and the minimum required crowd size of each performance.
- $2 \cdot m$ lines, each two consecutive lines describing the favourite bands of one of the m people:
  - One line with an integer k ($1 \leq k \leq n$), the number of the person's favourite bands.
  - One line with k distinct integers $b_1, \ldots, b_k$ ($1 \leq b_i \leq n$), the person's favourite bands.

It is guaranteed that the sum over the number of favourite bands is at most $5 \cdot 10^5$.

Output

If it is impossible to avoid a flop, output "`impossible`". Otherwise, output "`possible`". If it is possible to avoid a flop, output in addition the number of bands in the chosen lineup, followed by the bands in that lineup in any order.

If there are multiple lineups that avoid a flop, you may output any one of them.

Sample Input 1	**Sample Output 1**
3 3 2 1 1 2 1 2 3 3 2 1	possible 2 1 2

In the first sample, the lineup consisting only of band 1 will also be accepted.

Sample Input 2	**Sample Output 2**
3 2 2 1 1 3 1 2 3	impossible

B Bye Bye Bilbo

Time limit: 1s

Bilbo Baggins is finally leaving the Shire! To celebrate his departure, Gandalf plans to set up a spectacular display of fireworks across the land.

The Shire has n hobbit holes numbered 1 to n, each of which is occupied by exactly one hobbit. Every hole $i > 1$ is connected by a lane directly to a unique hole p_i, with $p_i < i$, so that following the lanes always leads towards hole 1, the town.

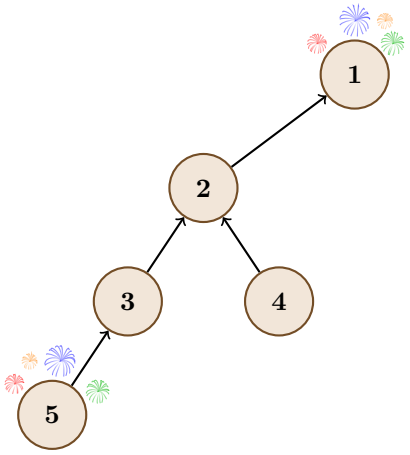


Figure B.1: Illustration of the first sample.

Gandalf must place fireworks at a minimum number of hobbit holes so that every hobbit can see at least one firework. The fireworks have a visibility range k. A hobbit at hole i can see a firework placed at hole j if it is placed at their own hole, or if j lies on the path from hole i to the town and the number of lanes between i and j is strictly less than k.

Input

The input consists of:

- One line with two integers n and k ($2 \leq n \leq 10^5$, $1 \leq k \leq n$), the number of hobbit holes and the firework range.
- One line with $n-1$ integers $p_2, p_3, \ldots, p_n$ ($1 \leq p_i \leq i-1$), where p_i is the next hole reached when going from hole i toward the town.

Output

Output one integer m, the minimum number of hobbit holes where a firework must be placed, followed by m distinct integers, the indices of those holes.

If there are multiple ways to place the fireworks using the minimum number of holes, you may output any one of them.

Sample Input 1	**Sample Output 1**
5 3 1 2 2 3	2 1 5

Sample Input 2	**Sample Output 2**
6 3 1 2 3 4 3	2 1 3

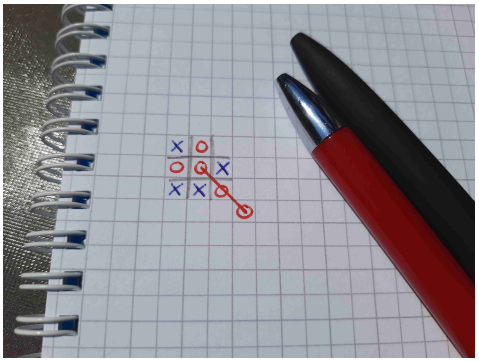
In the first sample, a firework at hole 1, i.e. the town, can only be seen by the hobbits at holes 1, 2, 3, and 4. A second firework at hole 5 can only be seen by the hobbit at hole 5.

In the second sample, a firework at hole 1 can only be seen by the hobbits at holes 1, 2, and 3. A second firework at hole 3 can only be seen by the hobbits at holes 3, 4, 5, and 6. So every hobbit can see at least one firework.

C Crosses and Circles

Time limit: 1s

Your friend watched some video about Connect Four and memorized the winning strategy for the starting player. What a loser! Whatever they are saying sounds about as relevant to your life as this paragraph is to the problem statement. Apparently, someone found a weak solution to Connect Four which can be compressed to just 150 kB and requires no game tree search.



An unusual game of tic-tac-toe.

To get them off their high horse, you challenge them to a real game. You pull out a piece of A4 graph paper and explain: "We will take turns picking any unpicked cell. The first player to get three consecutive cells in any row, column, or diagonal wins. I will start." Your friend is barely listening, but this is not just regular tic-tac-toe: you are playing on the whole paper. Convert your winning advantage.

Interaction

This is an interactive problem. Your submission will be run against an *interactor*, which reads from the standard output of your submission and writes to the standard input of your submission. This interaction needs to follow a specific protocol:

There is no initial input as you make the first move.

On each of your turns, print one line with two integers r and c ($1 \le r \le 59$, $1 \le c \le 42$), the row and column of the cell you pick. The cell must not have been picked before.

After each of your moves, the interactor will respond with one line with two integers r' and c' ($0 \le r' \le 59$, $0 \le c' \le 42$).

- If $r' = c' = 0$, the interaction is over. This happens if you won on your last move, made an illegal move, or your friend can win with their next move. Your program should terminate immediately.
- Otherwise, $1 \le r' \le 59$ and $1 \le c' \le 42$. This means that your friend picked the cell in row r' and column c'. It is guaranteed that neither player has picked this cell before. It is now your turn again, and the interaction continues from there.

The interactor is adaptive, so your friend's moves will depend on all previous moves.

Make sure you **flush** the standard output after every output. For example, you can use `fflush(stdout)` in C++, `System.out.flush()` in Java, `sys.stdout.flush()` in Python, `std::io::stdout().flush()` in Rust, and `hFlush stdout` in Haskell.

A *testing tool* is provided to help you develop your solution.

Read	**Sample Interaction 1**	**Write**
		9 7
10 6		
		9 6
9 8		
		8 7
10 7		
		10 8
8 6		
		11 9
0 0		

D Delphi Danger

Time limit: 3s

The temple of Apollon in Delphi. CC BY-SA 3.0 by Inkey on Wikimedia Commons

Have you ever been walking towards someone, dodged left only to see them do the same, and then entered an awkward shuffle to pass each other? Recently, a paper published at the Greek Commerce and Procurement Conference suggested that this may have already been a common problem in ancient Greece.

Back then, there were n merchant caravans that occasionally passed through the Thermopylae. When two caravans met while travelling in opposite directions, both would simultaneously choose to steer either cliffside or seaside. If they made different choices, they could safely pass each other. However, if both chose the same side, they had another opportunity to choose, repeating this process until they could safely pass. Caravans were unable to recognize each other from a distance and would each follow a deterministic strategy to decide which way to steer. Since steering seaside is much more dangerous, we say that the *risk* of a given strategy is defined as the total number of times it dictates to steer seaside. For example, a caravan may follow a strategy described by the sequence "cliffside, seaside, seaside, cliffside, seaside" and then only "cliffside" from then on. This strategy has a risk of 3.

The caravans convened at the oracle in Delphi to ask for advice on which strategies they should each adopt, but this went about as well as could be expected. Rather than helping, the oracle rattled off m prophecies of the following form: When caravans u and v meet, they will awkwardly steer in the same direction exactly t times before safely passing each other by steering in opposite directions on their $(t+1)$th attempt.

The real strategies were lost to history, but you wonder how risky they must have been. You think it is unwise to go against the oracle (this has ended poorly in the past) and want to minimize the *overall risk*, defined as the sum of the risks of all individual strategies.

Input

The input consists of:

- One line with two integers n and m ($2 \leq n \leq 2 \cdot 10^5$, $1 \leq m \leq 4 \cdot 10^5$), the number of caravans and the number of prophecies.
- m lines, each with three integers u, v, and t ($1 \leq u, v \leq n$, $u \neq v$, $0 \leq t \leq 10^9$), describing a prophecy: the strategies of caravans u and v will cause them to steer in the same direction exactly t times, and then steer in opposite directions.

It is guaranteed that each pair of caravans appears at most once.

Output

If there is no set of strategies that can fulfil all the prophecies, output "`impossible`". Otherwise, output "`possible`" followed by the minimum overall risk among all sets of strategies that fulfil the prophecies.

Sample Input 1	**Sample Output 1**

`5 3`
`1 2 2`
`1 3 0`
`4 5 2`
	`possible`
	`3`

Sample Input 2	**Sample Output 2**

`4 3`
`1 2 3`
`2 4 4`
`1 4 2`
	`impossible`

In the first sample, the strategies can be chosen as follows:

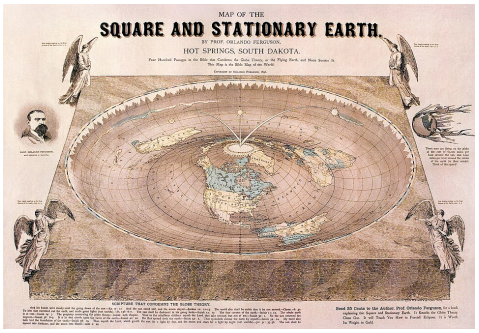
- caravan 1 always steers `cliffside` (risk 0)
- caravan 2 steers `cliffside`, `cliffside`, `seaside`, and then `cliffside` from then on (risk 1)
- caravan 3 steers `seaside` and then `cliffside` from then on (risk 1)
- caravan 4 always steers `cliffside` (risk 0)
- caravan 5 steers `cliffside`, `cliffside`, `seaside`, and then `cliffside` from then on (risk 1)

This satisfies all prophecies, yielding a minimum overall risk of 3.

E Egocentric Expedition

Time limit: 2s

You are the famous explorer *Christopherus Columb* and unlike most of your contemporaries you know that the earth and the world is flat and a perfect square. While others dreamfully deluded themselves with the fanatic heliocentric gospel that the earth orbits the sun, you in your superior wisdom discerned the higher truth of egocentrism: you alone remain, naturally, at the center of the world.



Map of the square and stationary earth. Public Domain by Prof. Orlando Ferguson on Wikimedia Commons

The only lifetime achievement remaining, worthy of your name, is to determine the exact area of the world. You cannot sail to the edge of the world yourself, as the world moves along with you such that you continue to be the center of the world. Therefore, you have to send your subordinates. They sail in a straight line direction chosen by you, starting at the center of the world. After their return, they will report the distance from you to the edge of the world in that direction.

As most crew members have been lured away by the heretic masses, there are only enough crew members left to operate a single ship. Knowing that there is barely enough time to send this ship out *twice*, are you wise enough to determine the area of the world?

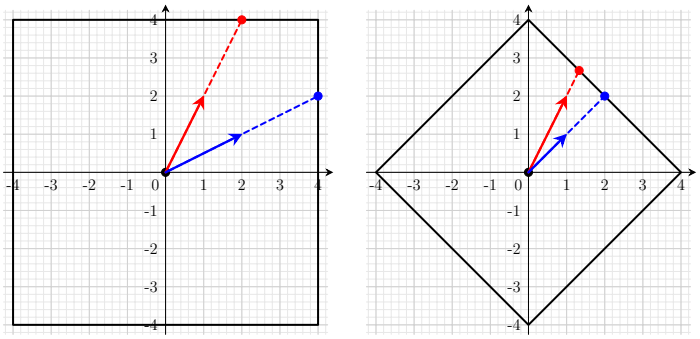


Figure E.1: Visualization of the two test cases used for the sample interaction. In the second sample (right), the ship is first sent in direction $(1, 2)$, where it finds the edge of the world at $\approx (1.33, 2.67)$, reporting distance ≈ 2.98. Then the ship is sent again in direction $(1, 1)$, finding the edge at $(2, 2)$, reporting distance ≈ 2.83.

Note that the rotation of the world square can be arbitrary, and the corner coordinates of the world do not have to be integer.

Interaction

This is an interactive problem. Your submission will be run against an *interactor*, which reads from the standard output of your submission and writes to the standard input of your submission. This interaction needs to follow a specific protocol:

The interactor first sends one line with an integer t ($1 \leq t \leq 10^4$), the number of test cases.

For each test case, you can ask *at most two* queries.

For every query, output "`? x y`" ($0 \leq |x|, |y| \leq 50, (x, y) \neq (0, 0)$), where x and y are integers, and (x, y) is the direction you send a ship to.

The interactor will respond to each query with one line containing a real value d ($0 < d \leq 50$), the distance the ship reported. This distance d is given with exactly 10 decimal places, and has an absolute *and* relative error of at most 10^{-7}.

After up to two queries, you should output "`! A`", where A is an integer, the area of the square. It is guaranteed that the square always has a positive integer area.

The interactor is not adaptive: the borders of the world are fixed before the first query.

Make sure you **flush** the standard output after every output. For example, you can use `fflush(stdout)` in C++, `System.out.flush()` in Java, `sys.stdout.flush()` in Python, `std::io::stdout().flush()` in Rust, and `hFlush stdout` in Haskell.

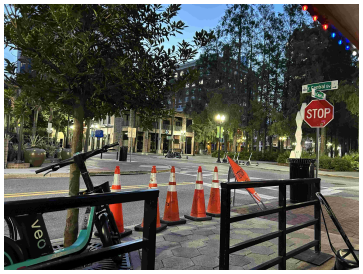
A *testing tool* is provided to help you develop your solution.

Sample Interaction 1

Read	Write
`2`	
	`? 1 2`
`4.4721359550`	
	`? 2 1`
`4.4721359550`	
	`! 64` `? 1 2`
`2.9814239700`	
	`? 1 1`
`2.8284271247`	
	`! 32`

F Fighting Fraud

Time limit: 1s



Picture of a GCPC courier. They are extremely discreet.

Got a job for you. This shady business called *Guaranteed Clandestine Private Couriers* is trying to hide something. Their public filings and tax records are spotless. As clean as a doctor's hands just before surgery. And you know what that means. They had dirt to scrub off, I'm sure of it. Got my hands on one of their drivers' delivery schedules. You know, a list of tasks to discreetly pick up and deliver an item. I bet we find something off with the schedule. Maybe an item that is picked up but never delivered. Maybe a delivery of something with no records where it came from. Heck, they might even list an item for some task long after the item has been delivered to keep the actual cargo off the books. Anyway, find out if the schedule is legit. If each item mentioned in the schedule is picked up and delivered exactly once, then the guy who cooks their books must be a genius. Guess they are paying him well. Speaking of, this one's double your regular fee. Half upfront for your discretion and the rest when I have your report. Don't call unless you're done!

Input

The input consists of:

- One line with an integer n, the length of the schedule ($1 \leq n \leq 10^5$).
- The next n lines each describe a schedule task, the ith of which contains:
  - The string "`pickup`" or "`dropoff`".
  - A string a_i consisting of English lowercase letters (`a-z`), the identifier of the item related to the ith task ($1 \leq |a_i| \leq 20$, $\sum |a_i| \leq 10^6$).

Output

Output "`yes`" if the schedule is valid and "`no`" otherwise.

Sample Input 1	Sample Output 1

`4`
`pickup apples`
`pickup oranges`
`dropoff oranges`
`dropoff apples`
	`yes`

Sample Input 2

```
6
pickup gcpc
pickup teams
dropoff gcpc
pickup jury
dropoff teams
dropoff gcpc
```

Sample Output 2

```
no
```

Sample Input 3

```
2
dropoff books
pickup books
```

Sample Output 3

```
no
```

Sample Input 4

```
4
pickup balloon
dropoff balloon
pickup balloon
dropoff balloon
```

Sample Output 4

```
no
```

Sample Input 5

```
1
dropoff fft
```

Sample Output 5

```
no
```

Sample Input 6

```
4
pickup timelimit
pickup correct
dropoff timelimit
dropoff correct
```

Sample Output 6

```
yes
```

G Garbled Garden

Time limit: 1s

After an extremely tiring day at work, Tessa arrived home to find her husband waiting proudly at the door. With a huge smile on his face, he announced that he had planted all her flowers in the garden. Grateful that her husband had taken some work off her shoulders, she went straight out to the garden. Her relief lasted exactly until she reached it.

While he did indeed manage to plant all of her flowers around the edge of the garden, unfortunately, the arrangement was very wrong. Roses stood where the orchids should have been. Hyacinths, dahlias, and chrysanthemums were scattered among the hydrangeas. The whole garden looked as though a toddler had planted them.

Tessa stared at the numbered tags still tied to the stems. She had arduously labelled every single flower with non-decreasing labels from left to right of the intended order. Flowers of the same species had matching tags since the order within one type of flower did not matter.

Her husband followed her gaze to the labels, shrugged, and said he thought that they were price tags and had not paid any attention to them ...

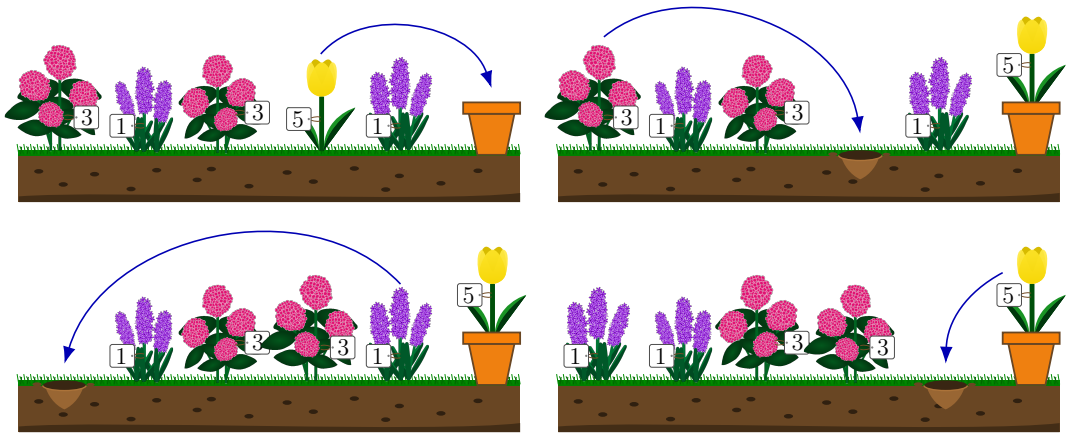


Figure G.1: Visualization of the second sample answer.

Sighing, she settled down and decided to fix this mess. Since the flowers were still very young and delicate, she cannot just dig out some of the flowers, rearrange them, and then plant them back. Instead, she can do the following operation:

- She chooses an integer m and a sequence of m distinct integers $p_1, \ldots, p_m$.
- First, she picks the flower at position p_1 and moves it back into a flowerpot.
- Then, she repeatedly chooses the flower at position p_i ($i \geq 2$), and moves it to the now empty position p_{i-1} of the previous flower.
- Finally, after $m-1$ iterations of the previous step, she finishes the operation by replanting the potted flower at the remaining empty position p_m.

Every flower can be used at most once within the *same* operation due to their fragile nature.

Determine the minimum number of operations Tessa has to do to fix the arrangement, and output one possible sequence of these operations.

Input

The input consists of:

- One line with an integer n ($2 \leq n \leq 5000$), the number of Tessa's flowers.
- One line with n integers $t_1, \ldots, t_n$ ($1 \leq t_i \leq n$ for each i), where t_i is the tag number of the flower at position i.

Output

First, output the minimum number of operations k ($0 \leq k \leq n$) to fix the arrangement. It can be proven that there always is an answer taking at most n operations.

Then, for each of the k operations, output:

- One integer m ($1 \leq m \leq n$), the number of flowers moved in this operation.
- m distinct integers $p_1, \ldots, p_m$ ($1 \leq p_i \leq n$ for each i), the positions of the flowers used in this operation. The order should match the order they are used in this operation.

Note that only the number of operations has to be minimal, not the number of flowers involved.

If there are multiple optimal solutions, you may output any one of them.

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

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

Sample Input 3	**Sample Output 3**
4 2 1 4 3	2 4 1 2 3 4 2 2 4

H Historical Hits

Time limit: 2s

Hitster is a tabletop game about placing songs into an ordered *timeline* by guessing their release year. As a longtime member of the *Companionless Playing Card Games (CPCG)* association, Harper has access to a single-player variant of Hitster. The rules are similar to regular Hitster. Harper starts with an empty timeline. In each round of the game, she draws a card which contains only a QR code on its front. She scans the QR code to listen to the song and afterwards guesses the release year of the song. Afterwards, she turns the card to reveal the song's true release year. Suppose that the true year is a and she guessed b. If Harper's timeline does not yet contain a song with a *true* release year that is strictly between a and b, she adds the card to her timeline.[1] Otherwise, she discards the card.

Harper just had a lengthy playthrough of the entire deck of n cards. Despite her thorough knowledge of music, her final timeline is shorter than she expected. Playing a second time now would be boring because Harper remembers the correct year of all the songs, along with her not-so-correct guesses.[2] Surely, she was just unlucky with the order in which she drew the cards, she thinks. To confirm her suspicion, Harper turns to probability theory. Suppose that the same cards are presented in an order drawn uniformly at random from the set of all $n!$ permutations. What is the expected length of Harper's final timeline if she repeats her guesses from her last playthrough for all of the n songs?

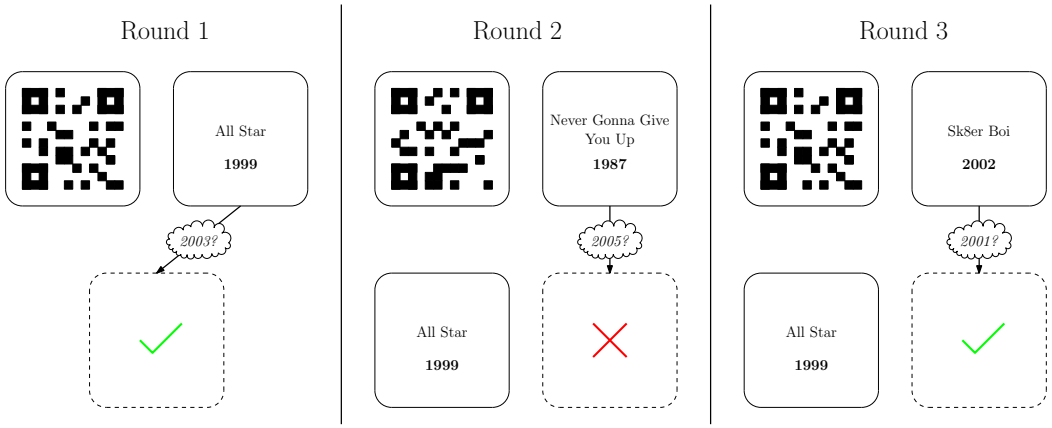


Figure H.1: Visualization of a playthrough of the game with the cards of Sample 3 presented in the order of the input. Harper's timeline is shown at the bottom. After the third round, her final timeline contains two cards. The top right of each round shows the back of the card, which Harper cannot see before making her guess.

Input

The input consists of:

[1] Her guess b never equals another card's true year, so the card's position in the timeline is uniquely determined.

[2] She developed this incredible memory through many long nights of playing a single-player variant of *Memory*.

- One line with an integer n ($1 \leq n \leq 3000$), the number of cards.
- n lines, the ith of which contains two integers a_i, b_i ($1 \leq a_i, b_i \leq 10^9$), the correct year and Harper's guessed year for the song of the ith card.

It is guaranteed that $a_i \neq a_j$ and $a_i \neq b_j$ for all $i \neq j$.

Output

Determine the expected length of Harper's timeline if the cards are presented in an order drawn uniformly at random from the set of all $n!$ permutations.

Let $M = 998\,244\,353$. The expected length can be represented as an irreducible fraction p/q where q is not divisible by M. Output one integer $p \cdot q^{-1} \bmod M$ (the unique integer x such that $0 \leq x < M$ and $x \cdot q \equiv p \bmod M$).

Sample Input 1	**Sample Output 1**
2 2002 2004 2001 2003	499122178

There are two possible orders. If the cards are drawn in the order of the input, then the first card is added to the timeline and the second card is discarded. If the cards are drawn in reverse order, then both cards are added to the timeline. Thus, the expected length is $3/2$. One can verify that $499122178 \cdot 2 \equiv 3 \bmod M$.

Sample Input 2	**Sample Output 2**
2 2001 2001 2002 2002	2

Sample Input 3	**Sample Output 3**
3 1999 2003 1987 2005 2002 2001	831870296

I Incremented Itinerary

Time limit: 2s

You recently started to go running in your free time. You got the shoes, you got the tracking app, and you even picked up your first injury a couple of days ago. Overall, you are really starting to like this new hobby of yours.

So far, your training routine was to run from your office to your home along the streets of your town. The town consists of n intersections, numbered from 1 to n, which are connected by m bidirectional streets of equal length. Your office is located at intersection 1, and your home is at intersection n.

As the computer scientist you are, you were of course following a shortest path. However, studies indicate that you should not just train to complete the same route in a faster time, but also to complete longer routes. Given that the last time you experimented with new training methods did not go so well, you want to take this new approach slowly. Therefore, you are looking for a route from your office to your home that is exactly one street longer than your current route. To keep the run interesting, you also want to avoid visiting any intersection more than once. Determine whether such a new training route exists.

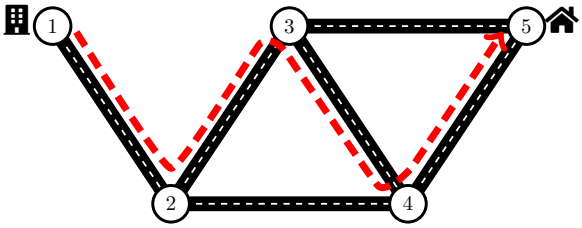


Figure I.1: Visualization of a possible new training route in the first sample.

Input

The input consists of:

- One line with two integers n and m ($2 \leq n \leq 10^5$, $1 \leq m \leq 2 \cdot 10^5$), the number of intersections and the number of streets.
- m lines, each with two integers a and b ($1 \leq a, b \leq n$, $a \neq b$), representing a bidirectional street between intersections a and b.

It is guaranteed that each pair of intersections is connected by at most one street. Further, it is guaranteed that there is at least one route from intersection 1 to intersection n.

Output

If there is a route that can serve as your new training route, output "`possible`". Otherwise, output "`impossible`".

Sample Input 1

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

Sample Output 1

```
possible
```

Sample Input 2

```
2 1
1 2
```

Sample Output 2

```
impossible
```

Sample Input 3

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

Sample Output 3

```
impossible
```

Sample Input 4

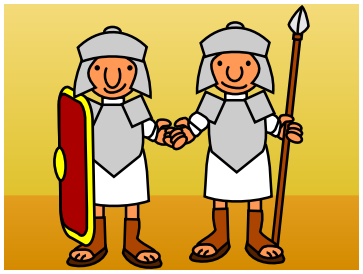
```
6 5
1 3
3 6
6 2
2 5
5 1
```

Sample Output 4

```
possible
```

J Junior Joining

Time limit: 2s



Patrol pair with shield and spear. Public Domain on publicdomainvectors (edited).

Arya, the Commander of Gil'ead, has received $2 \cdot n$ new recruits. Since the Gil'ead City Protectorate Council has mandated more patrols, she intends to pair them into n pairs to patrol the area. To maximize the effectiveness of a patrol pair in the modern era, it should consist of one shield bearer and one spear carrier.

Each recruit i has shield-defense skill d_i, and spear-attack skill a_i. While skilled usage of their respective tool is important, one should never neglect the effectiveness of strong synergy. Because of long-standing rivalries between cities, strong bonds have formed within each city. With enough discipline, Arya is sure that there will not be any bad blood within a pair, but if both are from the same city, they will synergize exceptionally well.

The total fighting strength of a pair is the sum of the shield-defense skill of one, the spear-attack skill of the other, and if both recruits are from the same city c, also a synergy bonus of c.

Determine the maximum sum of fighting strength over all pairs that Arya can achieve with optimal pairing.

Input

The input consists of:

- One line with an integer n ($1 \leq n \leq 10^5$), half the number of new recruits.
- $2 \cdot n$ lines, the ith of which contains three integers d_i, a_i, and c_i ($1 \leq d_i, a_i, c_i \leq 10^6$), the shield-defense skill, spear-attack skill, and the home city of the ith recruit.

Output

Output the maximum achievable sum of fighting strength over all pairs.

Sample Input 1	Sample Output 1
1 4 2 1 3 2 2	6

There are only two recruits so they are paired together. Recruit 1 is assigned the shield, recruit 2 the spear. As they are not from the same city, only their skill is counted $4 + 2 = 6$.

Sample Input 2	**Sample Output 2**
2 1 2 5 4 2 5 5 1 5 3 2 1	18

Recruit 3 (shield) is paired with recruit 1 (spear). As they are both from city 5, they have 5 bonus strength from synergy ($5 + 2 + 5 = 12$ total). Recruit 2 (shield) and recruit 4 (spear) are paired, resulting in a fighting strength of 6.

Sample Input 3	**Sample Output 3**
2 1 7 2 1 5 2 8 1 6 7 1 6	27

One possible pairing is the following: recruit 3 (shield) with 1 (spear), and recruit 4 (shield) with 2 (spear).

K Keeping Cows

Time limit: 1s

Farmer Kiki has 42 cows. Their field can be thought of as a grid where each cell is a square with a side length of one metre. According to strict regulations of the German Cow Perimeter Conglomerate, the cows must graze on a fenced area that is exactly a square metres large.

A cow. CC BY-SA 3.0 by Kim Hansen on Wikimedia Commons

Unfortunately, Farmer Kiki's supplier messed up their order of fence pieces. Instead of mostly straight fence pieces and just a few corner fence pieces, they only received an enormous amount of corner fence pieces. There are even more pieces than would fit on the field. One corner fence looks like this:

```
....
.○#.
.#..
....
```

Here '○' denotes the center of the fence, '#' denotes the sides of the fence, and '.' denotes empty space.

Farmer Kiki can rotate and move each corner fence grid-aligned. How may they place them such that there is a side-connected component of exactly a empty cells that is surrounded by fence pieces? More exactly, for each interior cell of that component all four orthogonal neighbours have to either also be an interior empty cell or part of a fence piece.

Farmer Kiki's field is 100×100 cells large and they cannot place any fence piece outside this area.

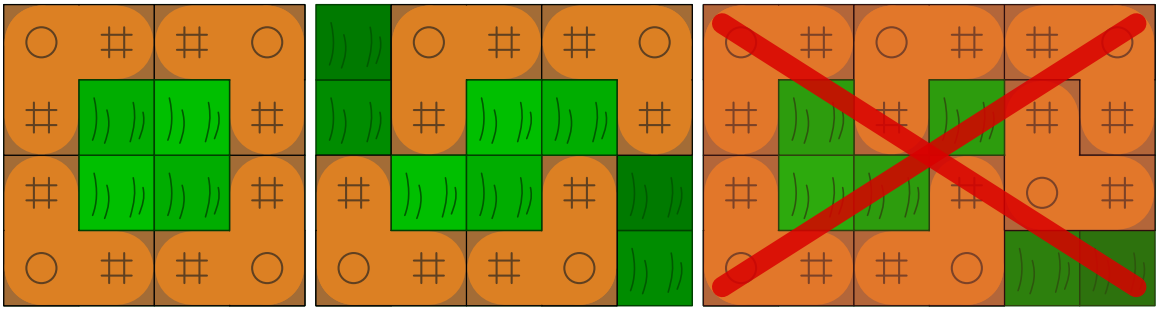


Figure K.1: Two valid and one invalid construction for the first sample with fenced area 4. The left construction represents the given sample answer. On the right, the four interior cells are not side-connected.

Input

The input consists of:

- One line with an integer a ($1 \leq a \leq 5000$), the exact number of cells the cow enclosure should contain.

Output

Output two integers h and w ($1 \leq h, w \leq 100$), the height and width of the used area. Then output the fence placement represented by h strings of length w containing only '`#`', '`.`', and '`O`' (the letter, not the digit '`0`').

If there are multiple valid solutions, you may output any one of them.

Sample Input 1

```
4
```

Sample Output 1

```
4 4
O##O
#..#
#..#
O##O
```

Sample Input 2

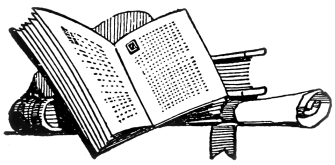
```
9
```

Sample Output 2

```
7 9
O##OO##.#
#..###O#O
#......#.
O##O.##O.
#..##O...
O##O.....
...#.....
```

L Lyrical Leisure

Time limit: 1s



Ornament from a 1923 magazine. Public domain on Wikimedia Commons by an anonymous author.

Hannah's favourite leisure activity is to write poems. She particularly likes to write about the small things in everyday life and about neat patterns. For example, she enjoys to write poems utilizing palindromes, since her name is also one! A palindrome is a string which is equal to its reverse, ignoring the letter cases.

However, she would like to introduce a small twist. In the grocery store, she was in the fruit section. When she was about to take a banana, she realized the word consists of the letter '`b`', followed by the palindrome '`anana`' of length 5. Her whole life, she has played around with palindromes, but never with words containing them.

At home, she immediately started to think of interesting words containing palindromes. Given two lengths n and k, she would like to find a method that constructs a string of length n, whose longest palindromic substring is of length k. A substring of a string s is a sequence of consecutive characters of s.

For now, she does not care about whether the string this method provides is an actual word in the dictionary, and any string will suffice.

Input

The input consists of:

- One line with two integers n and k ($1 \leq k \leq n \leq 5000$), the length of the string and the length of the longest palindromic substring.

Output

Output a string of length n consisting of English letters (`a-z` and `A-Z`) whose longest palindromic substring is of length k.

If there are multiple valid solutions, you may output any one of them.

Sample Input 1	Sample Output 1
`6 5`	`banana`

Sample Input 2	Sample Output 2
`4 1`	`abca`

Sample Input 3	Sample Output 3
`6 6`	`Hannah`

Sample Input 4	**Sample Output 4**
4 3	GCPC

M Mirror Magic

Time limit: 2s

Mia and Mark both own a chandelier, each of which has n candleholders. After the installation, Mia wonders whether she can hide a part of the room behind a vertically placed mirror for a magic trick. Of course, the mirror should be placed such that Mark would not notice its existence. She believes that she can ensure that Mark will not be able to see her or himself in the mirror, and that she can use clever lighting to hide any possible weirdness resulting from mirroring the room's walls. What worries her most are the chandeliers. Mark knows the precise positions of all the candleholders and would immediately notice if the chandelier in the mirror looked different from the chandelier behind the mirror he expects to see. Naturally, all candleholders of one chandelier should lie on one side of the mirror, and all candleholders of the other chandelier on the other side.

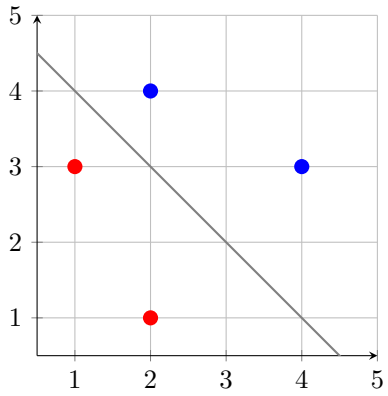


Figure M.1: Illustration of the third sample.

Input

The input consists of:

- One line with an integer n ($1 \leq n \leq 10^5$), the number of candleholders in each chandelier.
- n lines, each containing two integers x_i and y_i ($-10^6 \leq x_i, y_i \leq 10^6$), the coordinates of the ith candleholder of Mia's chandelier.
- n lines, each containing two integers x_i and y_i ($-10^6 \leq x_i, y_i \leq 10^6$), the coordinates of the ith candleholder of Mark's chandelier.

It is guaranteed that all $2 \cdot n$ points are pairwise distinct.

Output

Output "`possible`" if it is possible to place the mirror as desired, and "`impossible`" otherwise.

Sample Input 1

```
1
0 0
1 1
```

Sample Output 1

```
possible
```

Sample Input 2

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

Sample Output 2

```
impossible
```

Sample Input 3

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

Sample Output 3

```
possible
```

Sample Input 4

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

Sample Output 4

```
possible
```

Sample Input 5

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

Sample Output 5

```
impossible
```

Sample Input 6

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

Sample Output 6

```
impossible
```

Sample Input 7	**Sample Output 7**
3 -1 -1 -2 -2 -2 1 0 1 1 -1 1 2	impossible