Problem A. Gratitude

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Ben heard about studies by Emmons and McCullough that suggest that intentionally practicing gratitude has a lasting effect on people's happiness. Since he wants to be happy too, he decided that at the end of each day he will think back over the past day and write down three things he is thankful for, one thing per line. At the end of N days in which he practiced this exercise, he was curious to know which things appear the most on his list. Help Ben get the K things he was grateful for most frequently.

Input

The input begins with one line containing two space-separated integers, N and K, in that order. Then follow $3N$ lines containing Ben's notes from N days. You may assume that the three lines that correspond to the same day contain no repetitions. That is, if you partition the input into N chunks of 3 consecutive lines, no chunk contains two identical lines.

Limits

- $1 \leq K \leq 3N \leq 100\,000$
- Each input line contains at most 50 (ASCII) characters.

Output

The output should represent the list of things that Ben is grateful for, ordered by frequency of appearance in Ben's list (with the most frequent item first). In case of two items with equal frequency, the most recent item should appear first. That is, in case of a tie in the number of appearances, the item whose last appearance is later in the input should appear earlier in the output. Finally, if there are more than K different items in Ben's list, your output should contain only the K first items (according to the required order).

Examples

standard input	standard output
2 2 Supportive parents Being able to solve a hard problem Good food Fun game with friends Good food Being healthy	Good food Being healthy
2 6 Supportive parents Being able to solve a hard problem Good food Fun game with friends Good food Being healthy	Good food Being healthy Fun game with friends Being able to solve a hard problem Supportive parents

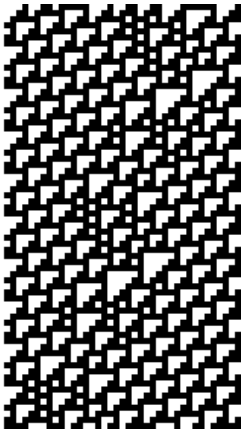
Note

Sample Explanation 1 `Good food` is the only item that appears twice in Ben's list, so it should appear first in the output. All other items appear once in the input, but `Being healthy` takes precedence as it is the most recent.

Sample Explanation 1 Here there are only 5 different items that Ben is grateful for, so there are only 5 lines of output. In this list, `Good food` is first in the output since it appears twice in the input, and the other items are ordered by last appearance in Ben's list.

Problem B. Rule 110

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes



Ann is decorating her office with the coolest arrangement of lights ever. She is using very long LED strips, where each individual cell is switched on or off every second, according to the following simple and pretty algorithm. At each step, the status of each cell (0 for off and 1 for on) is determined from the status of its two neighbor cells on the strip (left and right) and its own status, according to the following table:

Current pattern	111	110	101	100	011	010	001	000
New state for center cell	0	1	1	0	1	1	1	0

Ann is choosing an initial configuration for the cells and she marvels at the resulting animation, which happens to be highly similar to Conway's Game of Life, with interesting behavior on the boundary between stability and chaos.

Input

The input is composed of two lines.

- The first line contains the initial configuration, as a string of 16 characters `0` and `1`. All the cells to the left and to the right of this string are considered to be 0.
- The second line contains the number N of steps to perform.

Limits

- $0 \leq N < 2^{60}$
- The LED strip is considered to be large enough to ensure that no 1-cells will ever reach the ends of the strip.

Output

The output should contain a single line with a single integer that is the total number of 1-cells in the final configuration.

Example

standard input	standard output
0001001101111100 5	11

Note

Sample Explanation the output is 11 since we have the following five steps:

```
...00000000100111011111000...
...00000001101111110001000...
...00000011111000100111000...
...00000110001001101111000...
...00001110011011111101000...
...00011010111111000111000...
```

where everything not displayed contains only 0-cells.

Problem C. Safe Distance

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

The past year has been difficult, with a virus spreading among the population. Fortunately, Alice knows that one of the keys to be healthy is to keep a safe distance from other people.

Alice is currently in a closed room, represented in the $2D$ plane, with width X and height Y. There are N other people inside the room, and we're given their (x_i, y_i) coordinates.

We consider Alice and the N people as points in the $2D$ plane. Alice's initial position is $(0, 0)$ and she wants to move to the exit at position (X, Y). She can move freely in any direction inside the room, but can not step outside the room bounds.

Find the maximum distance Alice can keep from other people while moving from $(0, 0)$ to (X, Y).

Input

The input begins with one line containing two space-separated integers, X and Y, where X is the width, and Y is the height of the room. The second line consists of a single integer N, the number of people in the room. Then N lines follow, each of them consisting of two floating-point numbers x_i and y_i, the coordinates of the i^{th} person in the room.

Limits

- $1 \leq X, Y \leq 1\,000\,000$
- $1 \leq N \leq 1\,000$
- $0 \leq x_i \leq X$
- $0 \leq y_i \leq Y$

Output

The output consists of a single value d, the maximum safe distance, as a floating-point number.

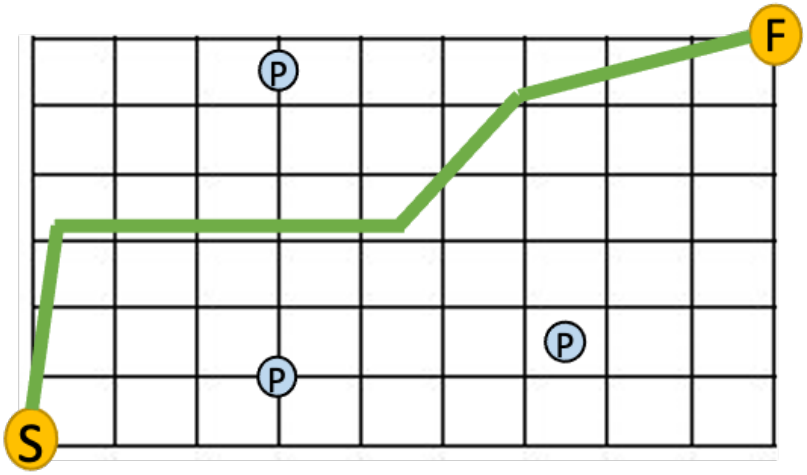
An additive or multiplicative error of 10^{-5} is tolerated: if d is the answer, any number either within $[d - 10^{-5}; d + 10^{-5}]$ or within $[(1 - 10^{-5})d; (1 + 10^{-5})d]$ is accepted.

Example

standard input	standard output
8 6 3 3 1 3 5.5 6.5 1.5	2.250000

Note

Alice can keep a distance of 2.25 from every other person, and this is the best she can do. The picture below shows a possible path (in green).



Problem D. Jogging

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Pheobe has heard that exercise has a tremendous affect on both physical and mental health. She never went jogging before, and she wants to try it out. However, she knows that she gets bored quickly and it is difficult for her to repeatedly do the same thing. In order to get into the habit of jogging, Pheobe decided to take up a challenge: she will go out for a run every evening as long as she finds an interesting path to take. For her, a path is interesting if it goes through a street where she did not run before. Pheobe asks for your help in understanding what is the maximum number of days she can run if she plans well.

Pheobe gives you as input a description of her neighborhood. She lives on an intersection, and she describes all of the intersections in the neighborhood. She also tells you which intersections are connected by streets, and what is the length of each street in meters. Every street connects two different intersections, and it is not possible that two different streets connect the same two intersections. In addition, you may assume that Pheobe only describes streets that can be reached from her home and that streets can be traversed in both directions as Phoebe is on foot.

A valid run starts and ends in Pheobe's home, and its length should be within the range that Pheobe specifies. When Pheobe enters a street, she does not have to go through the entire street (she is allowed to turn around at any point), but even if she does that, it counts as if she has seen the entire street for the purpose of determining whether runs are interesting. A run is considered interesting if it includes a street (or a segment of it) that did not appear on previous runs. Reaching an intersection does not count as visiting all streets adjacent to it.

Input

The input begins with one line containing four space-separated integers, I S L U, in that order. I represents the number of intersections in the neighborhood, and S represents the number of streets. L is the minimum number of meters in a valid run, and U is the maximum number of meters in a valid run.

Then, follow S lines, each line representing a street. Each such line contains 3 space-separated integers, i j ℓ, in that order. Integers i and j are the intersections that the street connects, and ℓ is the length of the street in meters. The intersections are numbered between 0 and $I-1$ such that Pheobe lives in

intersection number 0.

Limits

- $1 \leq I \leq 100\,000$
- $0 \leq S \leq 100\,000$
- $1 \leq L \leq U \leq 42\,195$ (Pheobe will not run more than a marathon)
- $1 \leq \ell \leq 1\,000$

Output

A single line containing a single integer holding the length of the longest sequence of interesting runs.

Examples

standard input	standard output
4 4 80 90 0 1 40 0 2 50 1 2 30 2 3 10	3
2 1 7 7 0 1 3	1

Note

Sample Explanation 1 Here is an example for an interesting 3-day jogging plan for the first sample input:

- On the first day, run back and forth on the street between 0 and 1 (80 meters).
- On the second day, run for 40 meters on the street to 2 and then go back the same way (80 meters).
- On the third day, run on the street to 1, then run for 5 meters in the direction of 2, and then go back the same way (90 meters).

Sample Explanation 2 Here is one possible valid run: Run from 0 to 1, then back to 0, then half a meter in the direction of 1, and then back to 0.

Problem E. Cakes

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

This summer, you plan to organize a large party and invite many friends. They have a sweet tooth, so you plan to bake nice cakes for them. You know the recipe for a nice chocolate cake, and you want to cook as many of them as possible.

Given the N ingredients needed to make a single cake and the ingredients that you have in your kitchen, how many cakes can you make?

Input

- The first line of the input contains a single integer N.
- Then, N lines follow, one for each ingredient. Each of these lines contains two positive integers: the first one is the required quantity of this ingredient per cake, the second one is the quantity of this ingredient you have in your kitchen.

textbfLimits

- $1 \leq N \leq 10$
- All ingredient quantities will be integers between 1 and 10 000.

Output

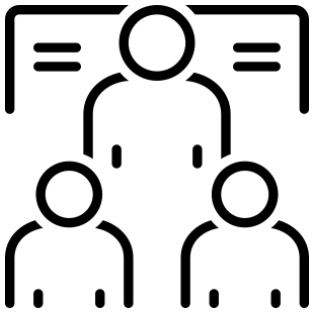
The output should contain a single integer: the maximum number of cakes you can make using the available ingredients.

Examples

standard input	standard output
3 100 500 2 5 70 1000	2
3 100 50 2 5 70 1000	0

Problem F. Mentors

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes



The Happy Tree Friends have gathered for their annual meeting, in which they take their most important decisions for the year to come. This year, they will set up a mentoring program to help each other take better care of their loved ones. This programme follows a tree-like hierarchical structure as follows.

The N members of the programme are ranked from 1 to N (each rank is assigned once), by increasing seniority. For the mentorship programme to be efficient, a person ranked A can mentor a person ranked B only if $A > B$. The most senior Happy Tree Friend can have no mentor, but everybody else has a unique mentor. Conversely, everybody is allowed to mentor from zero to two people.

However, Mr. Pickles, who was assigned the rank R, plans to take a sabbatical this year. Thus, he will not be able to mentor anybody, and the Happy Tree Friends should choose their hierarchical structure among those trees in which the node labelled R is a leaf.

Aiming to help his friends to choose such a tree, Mr. Pickles decides to first count how many trees would match his constraint. Unfortunately, he stopped school early, and thus did not learn how to manipulate integers of arbitrary size. Instead, he counts modulo M, where M is a fixed positive integer: this is already enough for most purposes in life.

What is the number L that Mr. Pickles will obtain after counting all suitable trees?

Input

The input consists of a single line, with three space-separated integers: R, N, M, in that order.

Limits

- $1 \leqslant R \leqslant N \leqslant 2021$
- $1 \leqslant M \leqslant 1\,000\,000\,000$

Output

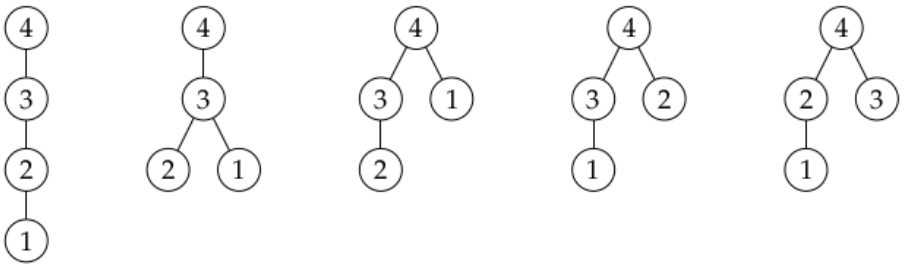
The output should contain a single line with the single integer L, which is the number of tree-like hierarchical structures that would match Mr. Pickles' constraints, counted modulo M.

Examples

standard input	standard output
2 4 2	1
2 4 3	0

Note

Sample Explanation 1 The node with label $R = 2$ is a leaf in exactly 3 of five trees listed below, and thus there are 3 trees that match Mr. Pickles' constraints. The only meaningful feature of our trees is parenthood, which represents mentorship relations, and thus there is no notion of left child or right child of a node. Mr. Pickles counts modulo $M = 2$, and therefore he ends up with the number $L = 3 \pmod 2 = 1$.



Sample Explanation 2 Mr. Pickles now counts modulo $M = 3$, and thus he ends up with the number $L = 3 \pmod 3 = 0$.

Problem G. Decoration

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	256 megabytes

After all these months of lockdown, you are tired of the interior decoration of your home and decide to redesign it. Hence, you read many blog posts and magazines about Feng Shui decorating and other recent trends on home design. After some time of thinking, you decide to reproduce the idea of the famous designer Sweta Marc for replacing your bookcase with a new one you will build.

According to S. Marc, a harmonious bookcase always has several shelves spaced in an heterogeneous manner, and always following some very precise rules. More precisely, such a bookcase has a serenity value N and is composed of $K+1$ shelves spaced by $s_1, \ldots, s_K$ millimeters between each other, from the bottom to the top. According to S. Marc ideals, these spaces should verify the following properties:

1. They should be heterogeneous, i.e., no two spaces have the same height.
2. They should be not too high, i.e., for all $i \in [1, K]$, we should have $0 \leq s_i < N$. Note that one of these spaces might actually have size 0: this is one of the oddities which make Sweta's tastes so visually attractive (arguably, this is a loss of space, but you are ready for that in the name of elegance, well-being... and trendiness).
3. They should be serene, i.e., for all $i \in [1, K-1]$, Sweta prefers if s_{i+1} is congruent modulo N to s_i plus the number of divisors of s_i. (Yes, Ms. Marc is sophisticated and loves arithmetic.)

You tried to design a bookcase according to the advice of Sweta Marc, but you find it hard to satisfy all the requirements. The only few solutions you found result in a bookcase which is too tall for your place.

Therefore, you decide to write a program which, given the number of shelves K and the serenity value N, computes the values of the spaces $s_1, \ldots, s_K$ of one of the minimum height bookcases, i.e. a bookcase where the sum of spaces $s_1 + \cdots + s_K$ is the smallest.

Input

The only line of input contains two integers N and K separated by a space.

Limits

- $1 \leq N \leq 1\,000\,000$

- $1 \leq K \leq 1\,000\,000$

Output

The output should contain a single line containing either:

- -1 if it is not possible to satisfy Sweta Marc's prescriptions for the given values of K and N,
- otherwise, K integers $s_1, \ldots, s_K$, corresponding to the spaces between the shelves of one of the minimum height bookcases satisfying the constraints. If several solutions are possible, the output should contain any of them.

Examples

standard input	standard output
18 10	11 13 15 1 2 4 7 9 12 0
168 9	1 2 4 7 9 12 18 24 32

Note

We recall the following mathematical definitions (a and b are arbitrary integers):

- a divides b if there exists an integer q such that $b = aq$;
- a is a divisor of b if $b \neq 0$ and a divides b;
- a is congruent to b if N divides $b - a$.

Problem H. Figurines

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Bob has a lot of mini figurines. He likes to display some of them on a shelf above his computer screen and he likes to regularly change which figurines appear. This ever-changing decoration is really enjoyable. Bob takes care of never adding the same mini figurine more than once. Bob has only N mini figurines and after N days he arrives at the point where each of the N figurines have been added and then removed from the shelf (which is thus empty).

Bob has a very good memory. He is able to remember which mini figurines were displayed on each of the past days. So Bob wants to run a little mental exercise to test its memory and computation ability. For this purpose, Bob numbers his figurines with the numbers $0, \ldots, N-1$ and selects a sequence of N integers $d_0 \ldots d_{N-1}$ all in the range $[0; N]$. Then, Bob computes a sequence $x_0, \ldots, x_N$ in the following way: $x_0 = 0$ and $x_{i+1} = (x_i + y_i) \bmod N$ where mod is the modulo operation and y_i is the number of figurines displayed on day d_i that have a number higher or equal to x_i. The result of Bob's computation is x_N.

More formally, if we note $S(i)$ the subset of $\{0, \ldots, N-1\}$ corresponding to figurines displayed on the shelf on day i, we have:

- $S(0)$ is the empty set;
- $S(i)$ is obtained from $S(i-1)$ by inserting and removing some elements.

Each element $0 \leq j < N$ is inserted and removed exactly once and thus, the last set $S(N)$ is also the empty set. The computation that Bob performs corresponds to the following program:

$$
\begin{array}{l}
x_0 \leftarrow 0 \\
\text{for } i \in [0; N-1] \\
\qquad x_{i+1} \leftarrow (x_i + \#\{y \in S(d_i) \text{ such that } y \geq x_i\}) \mod N \\
\text{output } x_N
\end{array}
$$

Bob asks you to verify his computation. For that he gives you the numbers he used during its computation (the $d_0, \ldots, d_{N-1}$) as well as the log of which figurines he added or removed every day. Note that a mini figurine added on day i and removed on day j is present on a day k when $i \leq k < j$. You should tell him the number that you found at the end of the computation.

Input

The input is composed of $2N + 1$ lines.

- The first line contains the integer N.
- Lines 2 to $N + 1$ describe the figurines added and removed. Line $i + 1$ contains space-separated `+`j or `-`j, with $0 \leq j < N$, to indicate that j is added or removed on day i. This line may be empty. A line may contain both `+`j and `-`j, in that order.
- Lines $N + 2$ to $2N + 1$ describe the sequence $d_0, \ldots, d_{N-1}$. Line $N + 2 + i$ contains the integer d_i with $0 \leq d_i \leq N$.

Limits

- $1 \leq N \leq 100\,000$

Output

The output should contain a single line with a single integer which is x_N.

Example

standard input	standard output
3 +0 +2 -0 +1 -1 -2 1 2 2	2

Note

Sample Explanation The output is 2 since

- first, $x \leftarrow 2$ since $S(1) = \{0, 2\}$ and $\#\{y \in S(1) \text{ such that } y \geq 0\} = 2$;
- then, $x \leftarrow 0$ since $S(2) = \{1, 2\}$ and $\#\{y \in S(2) \text{ such that } y \geq 2\} = 1$;
- and finally, $x \leftarrow 2$ since $S(2) = \{1, 2\}$ and $\#\{y \in S(2) \text{ such that } y \geq 0\} = 2$.

Problem I. Emails

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	256 megabytes

Ariadna's blog is filled with delicious recipes and sensible advice for a healthy and balanced lifestyle. Unsurprisingly, it has thus gathered an impressive number of readers. This reader base is now stable, and Ariadna feels that it would be useful for them to interact more and form a tighter community, one that is not solely anchored to the blog.

Ariadna knows that some of the readers are already friends or acquaintances, and therefore have each other's email addresses. She thinks that a good start for developing the community would be for everyone to have everyone else's email address, so that everyone would be able to reach out to the entire group. Since she knows her blog's readers also greatly enjoy doing things in a "decentralized"fashion, she therefore devises the following protocol, to be started on day D:

- Every day at 8am, everyone sends the current list of contacts in their address book to all of the contacts in their address book.
- Every day at 8pm, everyone updates their address book, adding any new received email addresses.

If a person does not need to do any update at 8pm, then the process is said to have *converged* for this person, and she will no longer need to continue sending emails over the next days.

You are a skillful hacker and you have managed to get access to all of the blog readers' address books. You would like to surprise and impress Ariadna by notifying her of whether or not the process she proposes will lead to everyone getting everyone else's address. Moreover, if the process is meant to succeed, you want to give her a good estimate of how many days it would take. More precisely, if the process succeeds, you can either give her:

- the number E of days (including the first day) elapsed until the last update takes place, or
- the number of days (including the first day) elapsed until the process has *converged* on everyone's side. Note that, according to Ariadna's definition, this is equal to E+1.

Input

The first line of the input contains two integers N and M, corresponding to the number of readers and respectively to the number of pairs of readers that initially have each other's email address. Readers are numbered from 1 to N.

The M following lines each contain two integers, i and j, meaning that readers i and j initially have each other's email address. Note that this means that both reader i has reader j's address and reader j has reader i's address.

Limits

- $2 \leq N \leq 100\,000$
- $1 \leq M \leq 100\,000$

Output

The output should contain a single integer equal to either:

- -1 if the process does not lead to everyone eventually having everyone else's email address, or
- the estimated necessary number of days, otherwise. Note that this number may be equal to 0.

Examples

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

Note

- We assume the reader base is stable, i.e. no reader leaves and no additional reader joins throughout the process.
- We assume that everyone knows their own email address; receiving one's own address is simply ignored.
- You do no have to be "consistent"in your answers across several tests cases, meaning that you can output the value E for one test case and $E+1$ for another.

Sample Explanation 1 The process proceeds as follows:

- On day D at 8am:
  - Reader 1 sends the address of reader 2 to reader 2.
  - Reader 2 sends the addresses of readers 1 and 3 to readers 1 and 3.
  - Reader 3 sends the addresses of readers 2 and 4 to readers 2 and 4.

  - Reader 4 sends the address of reader 3 to reader 3.

- On day D after the 8pm update:
  - Reader 1's address-book has been updated and contains the addresses of readers 2 and 3.
  - Reader 2's address-book has been updated and contains the addresses of readers 1, 3 and 4.
  - Reader 3's address-book has been updated and contains the addresses of readers 1, 2 and 4.
  - Reader 4's address-book has been updated and contains the addresses of readers 2 and 3.
- On day D+1 at 8am:
  - Reader 1 sends the addresses of readers 2 and 3 to readers 2 and 3.
  - Reader 2 sends the addresses of readers 1, 3 and 4 to readers 1, 3 and 4.
  - Reader 3 sends the addresses of readers 1, 2 and 4 to readers 1, 2 and 4.
  - Reader 4 sends the addresses of readers 2 and 3 to readers 2 and 3.
- On day D+1 after the 8pm update:
  - Reader 1's address-book has been updated and contains the addresses of readers 2, 3 and 4.
  - The process has converged for reader 2 since there is no update.
  - The process has converged for reader 3 since there is no update.
  - Reader 4's address-book has been updated and contains the addresses of readers 1, 2 and 3.
- On day D+2 at 8am:
  - Reader 1 sends the addresses of readers 2, 3 and 4 to readers 2, 3 and 4.
  - Reader 4 sends the addresses of readers 1, 2 and 3 to readers 1, 2 and 3.
- On day D+2 after the 8pm update:
  - The process has converged for reader 1 since there is no update.
  - The process has converged for reader 4 since there is no update.

The last update takes place on day D+1, after **2** elapsed days. The process has converged for everyone on day D+2, after **3** elapsed days. The sample output contains the former value, **2**. Outputting the latter value, **3**, is an equally correct alternative.

Problem J. Daisy's Mazes

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

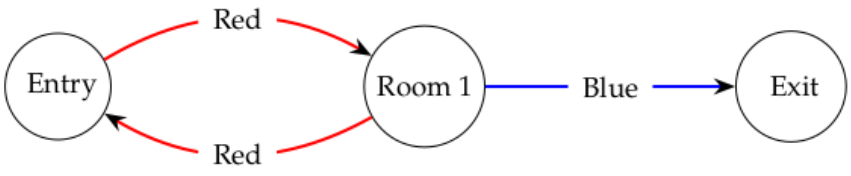
Daisy enjoys walking in mazes to evacuate the stress of a long day of work. The mazes that she likes are all composed of a set of rooms with one entry room, one exit room, and in each room there are several one-way doors leading to other rooms. Daisy's goal is to find a path from the entry to the exit.

Daisy has a technique to solve mazes. She has noticed that the different doors of any room have different colors and thus she can remember her path by keeping track of the colors of doors along her path. For that, she looks at the plan before entering the maze and builds a deck of colored cards corresponding to the colors of doors she needs to take. Whenever she enters a room, she goes through the door that has the color of the topmost card in her deck and then she discards this card.

It sometimes happens that Daisy's decks are "incomplete"and she arrives in a room with an empty deck or with a topmost card that has a color corresponding to none of the doors. In those cases, Daisy goes through one of the doors in the room and, instead of discarding the topmost card, she adds on top of her deck a card of the color of the door she took.

Let us consider the following example maze with three rooms and three doors: a red door from the entry to room 1, a second red door from room 1 back to the entry, and a blue door between room 1 and the exit. In this example maze (also depicted below) then:

- if Daisy starts with a deck containing a red card on top and a blue card below, she will go to room 1 and discard the red card, then go to the exit and discard the blue card;
- if Daisy starts with a deck containing a single red card then she will necessarily go to room 1 as a first step, discard the red card and from there she can choose to take the blue door and exit (it does not matter whether her deck is empty at the end) or she can choose to take the red door and goes back to her initial situation: in the entry room with a single red card;
- if she starts or arrives in the entry room with an empty deck, she will necessarily loop indefinitely. Indeed, the entry has only one door that leads to room 1. Once she arrives in room 1, her deck contains a red card on top and thus she has to take the red door and discard this card, which leads her back to the entry room with an empty deck.



Daisy knows that, in all of her labyrinths, she can always go from the entry room to the exit room with the right deck. However, some decks do not allow her to escape, whatever the choices she may ever do. She wonders: what is the minimal size of a deck that allows her to escape? Daisy gives you the plan of the maze and asks you to help her determine the minimal size of a deck that allows her go from the entry room to the exit if she makes the right choices.

Input

The first line contains three integers R, D, and C, separated by spaces. R is the number of rooms, D is the number of doors, and C is the number of colors. Rooms are numbered from 0 to $R-1$, and colors are numbered from 0 to $C-1$.

The next D lines each describes a door with three integers f, t and c, separated by spaces, and such that $0 \leq f \leq R-1$, $0 \leq t \leq R-1$, $f \neq t$, and $0 \leq c \leq C-1$. This indicates that there is a door from room f to room t, and that this door has color c.

Limits

- $2 \leq R \leq 50$
- $2 \leq D \leq 100$
- $2 \leq C \leq 20$

Output

The output should contain a single line with a single integer: the minimal integer S such that there is a deck composed of S cards that allows Daisy, if she makes the right choices, to go from the entry (the room numbered 0) to the exit (the room numbered $R-1$).

Examples

standard input	standard output
4 4 2 0 1 0 1 2 0 2 0 0 1 3 1	0
3 3 2 0 1 1 1 0 1 1 2 0	1

Note

Sample Explanation 1

- Daisy starts in room 0 with an empty deck
- She goes to room 1 with a card **0** on her deck
- She goes to room 2 with an empty deck
- She goes to room 0 with a card **0** on her deck
- She goes to room 1 with an empty deck
- She now has the choice to go the exit.

Sample Explanation 2 This example corresponds to the one given in the text with red represented as 1 and blue as 0.

Problem K. Unique Activities

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Emily is tired of having studied at home throughout 2020. She has noticed the same tasks occur over and over: she has to cook and wash the dishes. Then it's time for her class; afterwards she resumes washing the dishes, has to attend another class, washes some more dishes before cooking and washing the dishes for the last time of the day.

There is a part of her day she loves, though: it's when the sequence of activities she is currently carrying out happens only once during her day. She rejoices the most when that activity sequence is unique and really short.

Each activity is represented by an uppercase letter. Given the list of activities Emily has to carry out today, help Emily find the best moment of her day by finding the shortest substring that only occurs once in the input.

If Cooking is C, Dishes is D, and Studying is S, the list of activities in the example above are C D S D S D C D, and the shortest substring that occurs only once is D C. (All the one-letter substrings and the other two-letter substrings occur at least twice).

Input

The input consists of a single line, with a sequence of N uppercase letters (from 'A' to 'Z'). The line is terminated by a newline character which is not considered to be part of the input string.

Limits

- $0 < N \leq 300\,000$

Output

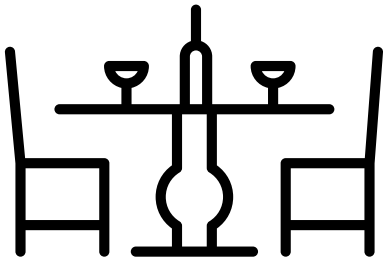
The output should contain a single line with the shortest substring that happens only once in the input string. If there are multiple shortest substrings (with the same length), output the one that occurs first.

Example

standard input	standard output
AABAABB	BA

Problem L. Restaurants

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes



Everybody is very happy to go back outside and to restaurants in Paris. However, for a while yet the restaurants have a very limited number of seats. We want to ensure that both restaurants can receive as many people as possible, and that customers go in their preferred seats.

We have N customers, numbered from 1 to N, and M restaurants, numbered from 1 to M. Each customer makes reservation in a subset of the restaurants, and give their list of reservations ordered by preference. Each restaurant ranks the reservations it received by some order of preference – for instance, the restaurant might wish customers who have signed up first to be ranked higher. Each restaurant i also has a capacity c_i, i.e. the maximal number of customers it can support.

Your task is to find an allocation of some of the customers in restaurants such that the following conditions are fulfilled:

1. No restaurant places more customers than their capacity.
2. Each customer is given a table in at most one restaurant.
3. There is no restaurant r and customer c having made a reservation for r, such that:
   - c has not been given a table or prefers r to the restaurant he was given a table in, and
   - r has some seats left or r is full but prefers c to at least one of the customers assigned to it.

Other remarks to note:

- Every customer has made at least one reservation.
- Restaurants only rank the customers having expressed a reservation for them. It is possible that a restaurant has no customers wishing to make a reservation.

Input

The first line contains N and M.

The M following lines describe capacities with the i-th line containing an integer c_i, the capacity of restaurant i.

N lines follow. The i-th line describes the list of reservations for customer i, sorted by preferences: the line contains a list of distinct space-separated integers (between 1 and M), from most to least preferred.

M lines follow. The i-th line describes the sorted preferences of restaurant i. This line contains either the number 0 when no customer made a reservation to restaurant i or it contains a list of space-separated distinct integers, the list of customers who made a reservation to restaurant i ordered from most to least preferred by the restaurant.

Limits

- $1 \leq N \leq 50\,000$
- $1 \leq M \leq 10\,000$
- total number of reservation options is at most $1\,000\,000$.
- $1 \leq c_i \leq N$

Output

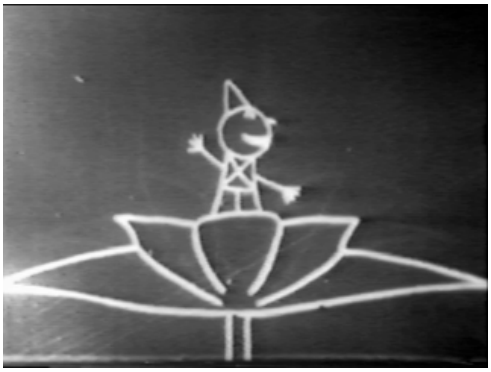
The output described the set of customers which have a table in one possible allocation (according to the rules above). The set is given with one customer per line, sorted ascending by id.

Example

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

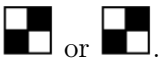
Problem M. Fantasmagorie

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



Émile's dreams often feature people or animals who become distorted into others. Émile would like to show the enchanting world of his dreams to everybody by creating the very first animated cartoon, in black and white of course. After waking up, Émile only remembers the initial and final forms appearing in his dream, not the transformation itself, so he's asking you to reproduce these "morphings"by detailing the steps taken for transforming the first image into the second.

The images in Émile's dreams are not just any black and white image, they respect the following three constraints. First, all pixels on the border of the image – the leftmost and righmost columns, and the top and the bottom rows – have the same color. Second, the image does not contain any 2×2 square of the form

 or .

The third constraint is more complex, and can be described as follows. Let us subdivide the image into regions, which are defined as the connected monochromatic areas of the image, i.e., they form the finest partition of the pixels such that any two adjacent pixels of the same color are in the same region. Two regions are considered adjacent if they contain adjacent pixels. In Émile's images, every region is adjacent to at most two other regions, and the region containing border pixels is adjacent to at most one other region.

You are given two black and white images of the same size $W \times H$, and your goal is to find a "morphing"transforming the first image into the second one. A morphing from image A to image B is a sequence of images starting with A and ending with B such that:

- each image (except the first) can be obtained from the previous one by flipping one bit;
- each image respects the three above constraints;
- the number of regions of each color does not change during the morphing.

Input

The first line of the input contains two space-separated integers: W and H. Then come H lines that describe the first image row by row, from top to bottom. Each of these lines is composed of W characters describing the row's W pixels, from left to right: the k-th character of the line is `'.'` if the k-th pixel of the row is white, and it is `'#'` if that pixel is black. Finally come H lines that describe the second image row by row, following the same format as above.

Output

If no morphing exists, the output should contain the word `"IMPOSSIBLE"` on a single line. Otherwise, the output should describe one possible morphing as follows: if the $(k+1)$-th image is obtained from the k-th image by flipping the pixel in column c and row r (with $0 \leq c < W$ and $0 \leq r < H$, where $c = 0$ represents the leftmost column and $r = 0$ represents the topmost row), the k-th output line contains the pair (c, r).

Limits and Remarks

- $1 \leq H \leq 64$
- $1 \leq W \leq 103$
- Émile's first and last images are distinct from each other.
- The output should contain at most 250 000 pixel flips.

Examples

standard input	standard output
4 3 .... .#.. ....#.	2 1 1 1
9 12 ....###... ...#.#... .#######. ...###... ...###... .#..#.... .#######. ...###.#. ...###... ..##.##.. ....#####.. ..##.##.. ..#...#.. ..#.#.#.. ..#...#.. ..#####.. ...#.#... ...#.#... ...#.#... .###.###.	IMPOSSIBLE