icpc International Collegiate Programming Contest

icpc.foundation

icpc Northwestern Europe

The 2025 ICPC Northwestern Europe Regional Contest

Problem Set

Northwestern Europe Regional Contest 2025

NWERC 2025

November 30, 2025

Karlsruhe Institute of Technology

Problems

Problem A
Arcade Crane
Time limit: 2 seconds

The local arcade just installed a new game, which is a new take on the classic claw machine. Inside the machine, there are n plushies arranged in a row, and above this row there is a mechanical claw which is operated by coins. For each coin inserted into the machine, the claw can be used once to grab exactly three consecutive plushies out of the row and then re-insert them somewhere in the row. The remaining plushies can always be pushed around (without changing their order) to make room for the re-insertion. The goal of the game is to sort the plushies by cuteness, and once they are sorted, the machine opens up and the person who achieves this wins *all* the plushies.

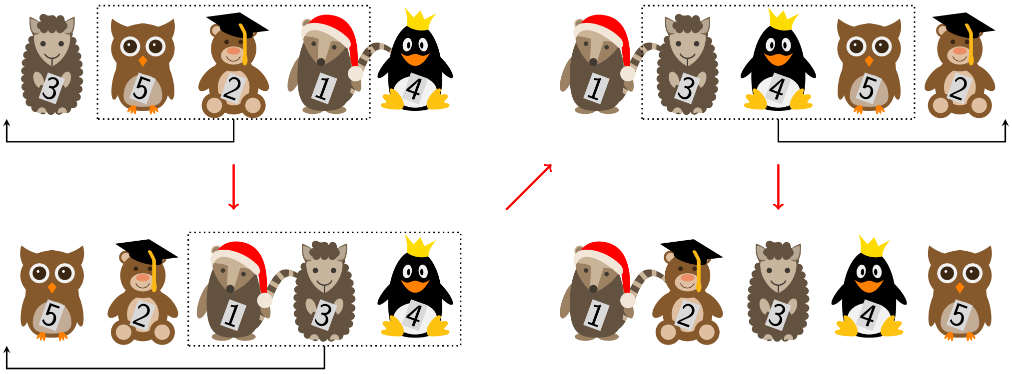


Figure A.1: Illustration of Sample Output 1.

Uli really wants to win the plushies, so they have done some research and found out that each plushie has a distinct cuteness value from 1 to n. To win, they need to sort the plushies in increasing order of these values. Equipped with the knowledge of all the cuteness values and a large stash of 5000 coins, how can they operate the machine to ensure they win the plushies?

Input

The input consists of:

- One line with an integer n ($5 \leq n \leq 1000$), the number of plushies.
- One line with n distinct integers $a_1, \ldots, a_n$ ($1 \leq a_i \leq n$ for each i), where a_i is the cuteness value of the ith plushie.

Output

First, output an integer q ($0 \leq q \leq 5000$), the number of operations. Then output q pairs of integers i and j ($1 \leq i, j \leq n-2$), describing the operations in order. The plushie positions in the machine are indexed from 1 to n. In an operation described by i and j, the plushies at positions i, $i+1$ and $i+2$ are grabbed and then re-inserted into the sequence such that they are in positions j, $j+1$ and $j+2$ after the operation.

It can be shown that a solution using at most 5000 operations always exists.

If there are multiple valid solutions, you may output any one of them. In particular, you do not need to minimize the number of operations.

Sample Input 1	Sample Output 1

<pre>5
3 5 2 1 4</pre>
	<pre>3
	2 1
	3 1
	2 3</pre>

Sample Input 2	Sample Output 2

<pre>6
6 5 4 3 2 1</pre>
	<pre>4
	1 3
	2 4
	3 4
	1 3</pre>

Sample Input 3	Sample Output 3

<pre>9
9 2 8 5 4 6 7 3 1</pre>
	<pre>5
	2 6
	5 1
	2 7
	5 3
	1 3</pre>

Sample Input 4	Sample Output 4

<pre>5
1 2 3 4 5</pre>
	<pre>1
	2 2</pre>

Note that $i = j$ is allowed (this does not change the sequence). For this test case, it would also be allowed to use no operations at all and just output "`0`".

Problem B
Bisecting Bargain
Time limit: 2 seconds

Illustration of Sample Output 4.
Photo by Jeroen Op de Beek

Emilia and Alex love Christmas markets. There are so many stalls to explore and so many delicious foods to try! The options seem endless: Schupfnudeln with Sauerkraut, langos, crêpes, roasted almonds, Bratwurst, twister fries and much more.

Here in Germany, some stalls still do not accept credit cards, so Emilia and Alex need to withdraw some cash from a nearby cash machine (also called an ATM, or Automated Teller Machine). Due to some fees, it is cheaper to withdraw all the cash they need at once, so they plan to do so.

This particular cash machine works in the following way: the user inputs an integer x, and then the machine chooses a selection of coins and banknotes that sums to €x. The cash machine can dispense all possible Euro coins and notes with values €1 and up: €1, €2, €5, €10, €20, €50, €100, €200, and €500. It has sufficiently many coins and notes of each type to form any combination that sums to x, and you do not know in advance which of these combinations it will dispense.

Since Emilia and Alex plan to visit some different stalls independently, they want to split the withdrawn amount evenly between them. While waiting in the long queue in front of the cash machine, Emilia suddenly realizes that this might not be possible, depending on which coins and banknotes the cash machine dispenses. For example, €42 might not be evenly splittable (see Sample 1), whereas no matter how €40 is dispensed by the cash machine, the cash can always be divided into two piles of €20.

Can the money always be split evenly if they withdraw €n from the cash machine?

Input

The input consists of:

- One line with an integer n ($1 \leq n \leq 10\,000$), the amount of cash in € that Emilia and Alex want to withdraw in total.

Output

If the money can always be split evenly, output "`splittable`". Otherwise, output the number of coins and notes, followed by their values, such that the values add to n and the money cannot be split evenly.

If there are multiple ways of choosing coins and notes that are not splittable, you may output any one of them.

Sample Input 1	Sample Output 1
`42`	`4` `10 20 10 2`

Sample Input 2	Sample Output 2
`40`	`splittable`

Sample Input 3	Sample Output 3
`5`	`1` `5`

Sample Input 4	Sample Output 4
`52`	`2` `2 50`

Problem C
Canal Crossing
Time limit: 4 seconds

Pixabay Content Licence by blaze_rob on Pixabay

To escape from the Christmas market hassle, you are planning a trip to Venice to see its beautiful canals and its many bridges. Unfortunately, you do not seem to be the only person with that plan, and Venice recently decided to charge for that pleasure. Therefore, you decide that crossing each bridge once is enough for you. Fortunately, every place can be reached from any other using only streets, without crossing any bridges. Interestingly, there is exactly one way to reach any other place using only streets.

After you gathered all this knowledge, now all that is left is to plan a trip that crosses each bridge exactly once. To keep things interesting, you also want to use each street at most once. What is the length of the shortest possible trip? Note that your tour can start at any place you choose, but it has to end where it starts.

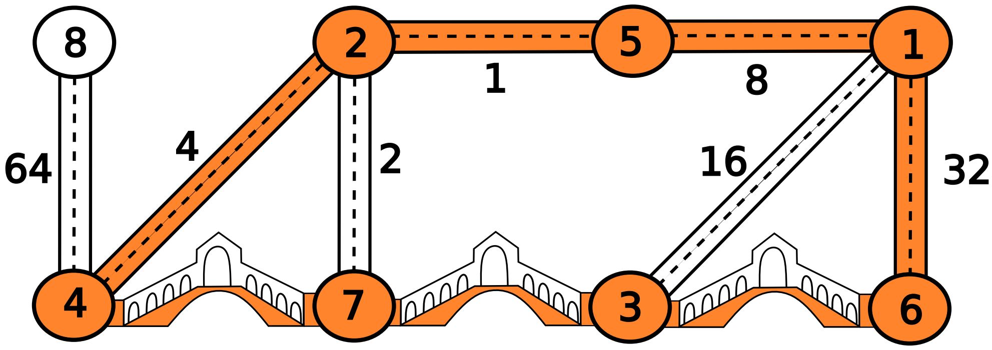


Figure C.1: Illustration of Sample Input 1 with a tour of length 45.

Input

The input consists of:

- One line with an integer n ($3 \le n \le 10^5$), the number of places in Venice.
- $n-1$ lines, each with three integers a, b, and w ($1 \le a, b \le n$, $a \ne b$, $1 \le w \le 10^6$), the starting and ending place of each street and its length.
- One line with an integer m ($1 \le m \le 5 \cdot 10^5$), the number of bridges in Venice.
- m lines, each with two integers a and b ($1 \le a, b \le n$, $a \ne b$), the places where each bridge starts and ends.

Bridges are short, and thus have length zero.

It is guaranteed that every place can be reached from any other place without using any bridges. Further, no bridge connects two places which are directly connected by a street, and all bridges are between distinct pairs of places. Lastly, it is guaranteed that at least one tour exists that crosses each bridge exactly once and uses every street at most once.

Output

Output the length of the shortest tour that crosses all bridges and uses each street at most once.

Sample Input 1	Sample Output 1
8 5 1 8 5 2 1 2 7 2 4 2 4 3 1 16 1 6 32 8 4 64 3 7 4 3 6 3 7	45

Sample Input 2	Sample Output 2
5 1 3 1 2 1 3 1 4 3 5 1 7 4 3 2 4 5 2 5 3 4	0

Problem D
Dreamcatcher
Time limit: 1 second

Lately, you are plagued by horrifying nightmares. In those dreams, a swarm of angry Fenwick bees chases you around a tree with heavy and light branches. You hate getting stung by their fiddling bits. Therefore, you decided to build a dreamcatcher three days ago, but the process is tricky, and you cannot figure out the details.

Planning your dreamcatcher became so stressful that you spent the last three nights awake, thinking about long strings of yarn, Carmichael numbers, and geometry. The only progress so far is some incomprehensible drawings (see Figure D.1) and the observation that the length of a chord that spans θ degrees of a circle with radius r is $2r \cdot \sin(\theta/2)$. But how is this going to help? You would rather have the dreams about Fenwick bees again than being tormented by this dreadful project for one more day. You need to solve this *now*!

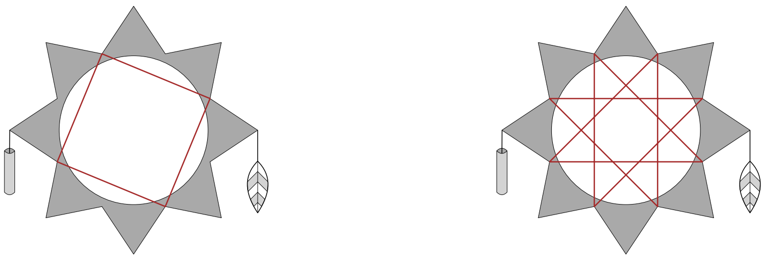


Figure D.1: A dreamcatcher with 8 notches, showing two ways of wrapping it in a string of yarn. The dreamcatcher on the left spans 2 notches per chord, while the dreamcatcher on the right spans 3 notches per chord.

To build a dreamcatcher, you take a wheel with n evenly spaced notches, numbered from 1 to n. You wrap a string of yarn around this wheel, spanning k notches per chord: starting at notch 1, you repeatedly connect the yarn k notches ahead until you reach notch 1 again. For example, with $n = 8$ and $k = 3$, you would go from notch 1 to 4, then 7, 2, 5, 8, 3, 6, 1.

A dreamcatcher's effectiveness depends on the amount of used yarn. You need to choose k, the number of notches to span per chord, such that the total length of yarn is maximized. The answer does not depend on the radius of the dreamcatcher.

Input

The input consists of:

- One line with an integer n ($3 \leq n \leq 10^9$), the number of notches on the wheel.

Output

Output an integer k ($1 \leq k < n$) that maximizes the length of used yarn when spanning k notches per chord.

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

Sample Input 1	Sample Output 1
8	3

Sample Input 2	Sample Output 2
5	2

Sample Input 3	Sample Output 3
6	5

Problem E
Erratic Lights
Time limit: 2 seconds

Your stall, before plugging in the lights. CC BY-SA 4.0 by Dietmar Rabich on Wikimedia Commons

You have almost finished setting up your stall for the Christmas market after many days of work. There is only one thing left to do: plugging in the string of lights. As you plug in the cord, you notice that you have brought the wrong lights – you planned to bring a string of blue lights, but instead, the lights are coloured in red, green, and blue! While leaning against your stall in despair, you accidentally bump into one of the light bulbs, and it immediately changes colour. You notice that every time you touch a light bulb, it randomly takes one of the three colours, each with equal probability.

The market starts in five hours already, and there is no way you have enough time to get this string of lights off the stall and install the one you wanted. At the very least, you want all the lights to have the same colour, no matter which colour that is. What is the minimal expected number of times you need to touch a light bulb to make all of them have the same colour?

For example, consider the first sample input and say you touch the red light until it turns blue. Every time you touch it, the new colour is either red, green, or blue. The expected number of times you need to touch it until it becomes blue is 3.

Input

The input consists of:

- One line with an integer n ($1 \leq n \leq 100$), the number of bulbs on the string of lights.
- One line with n characters, each character being either '`r`', '`g`', or '`b`', indicating the colours of the light bulbs at the start.

Output

Output the expected number of times you need to touch a light bulb to make all of them have the same colour, assuming an optimal strategy that tries to minimize the expected number of touches.

Your answer should have an absolute or relative error of at most 10^{-6}.

Sample Input 1	Sample Output 1
`2` `rb`	`3`

Sample Input 2	Sample Output 2
`3` `ggg`	`0`

Sample Input 3	Sample Output 3
5 rgbrg	7.5

Problem F
Fair Share
Time limit: 3 seconds

KIT teams having a celebration dinner after NWERC.
Photo by Christopher Weyand

Your university's teams are celebrating an outstanding performance at this year's NWERC in Karlsruhe. After a delicious dinner at a local restaurant, you call it a day. The trip home tomorrow will be a long one.

While trying to pay the bill, your group realizes that the restaurant takes no cash. Furthermore, it is too late to split the bill. Caught off-guard, everybody starts to open their wallets and puts some cash on the table. Someone has to pay the bill with their credit card and take the cash.

Each person i spent €b_i during the evening but has €a_i in cash to contribute to the group payment (if someone else pays). To keep it fair, the group does not want the person who pays the final bill (and takes the money from the cash pool) to end up paying more than their individual share. Thus, if person i is the one paying, then the remainder of the bill, after accounting for the cash contributions of the others, should not be more than their own share €b_i of the bill.

Help the group determine who should pay the final bill.

Input

The input consists of:

- One line with an integer n ($2 \leq n \leq 10^5$), the size of your dinner group.
- n lines, the ith of which contains two integers a_i and b_i ($1 \leq a_i, b_i \leq 1000$), the amount of cash person i would contribute if someone else pays and their share of the bill, respectively.

Output

If there is no suitable person to settle the bill, output "`impossible`". Otherwise, output one integer i ($1 \leq i \leq n$), the index of the person settling the bill.

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

Sample Input 1	Sample Output 1
3 4 3 5 4 1 3	3

Sample Input 2	Sample Output 2
5 1 4 8 1 1 4 2 5 4 6	impossible

The total bill is €20. If the first person settles the bill, they get €15 in cash from the others, paying €5 by themselves, which is more than their individual share of 4. A similar reasoning for each of the other people shows that they would also pay more than their individual share, and thus, there is no suitable person to settle the bill.

Sample Input 3	Sample Output 3
8 4 3 5 8 7 2 1 9 6 3 2 6 5 7 8 6	4

Problem G
Group Photo
Time limit: 4 seconds

The members of the No-Weather-too-Extreme Recreational Climbing society completed their 200th successful summit today! To commemorate the occasion, you will take a picture of all the members standing together in one row.

Illustration of Sample Input 1. Free Pexels License by PNW Production on Pexels

After the photo for the 100th summit eight years ago turned into a moderate fiasco, you are determined to get things right this time and make sure that people are arranged in an aesthetically pleasing way *before* taking the photo.

As this group is all about climbing mountains, you want the heights of the climbers in the photo to make the shape of a mountain. More precisely, the climbers should be arranged such that their heights are first increasing and then decreasing (the increasing or decreasing part is allowed to be empty). The heights of all climbers are pairwise distinct, so for simplicity we will say that the shortest climber has height 1, the second-shortest climber has height 2, and so on.

The climbers have already positioned themselves in a row in some arbitrary way that is not necessarily visually pleasing. You will select a subset of the climbers and rearrange their positions among themselves, with all other climbers staying exactly where they are. To keep chaos to a minimum, you want the number of moving climbers to be as small as possible.

What is the size of the smallest subset of climbers such that it is possible to reorder them so that the sequence of heights becomes first increasing and then decreasing?

Input

The input consists of:

- One line with an integer n ($1 \leq n \leq 5 \cdot 10^5$), the number of climbers.
- One line with n distinct integers $a_1, \ldots, a_n$ ($1 \leq a_i \leq n$ for each i), where a_i is the height of the ith climber in the current arrangement.

Output

Output the minimum number of climbers who need to move.

Sample Input 1	Sample Output 1
6 1 6 4 3 2 5	2

Swapping the first and the last climber is the optimal way to make the sequence of heights first increasing and then decreasing.

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

Sample Input 3	Sample Output 3
4 1 2 4 3	0

Sample Input 4	Sample Output 4
10 1 2 7 4 5 6 3 8 9 10	2

Problem H
Hasty Haul
Time limit: 3 seconds

Two teams participating in NWERC have hatched a devious plan: they want to get in the organizers' heads and mess with them a bit, but just subtly enough that they will not get caught.


The registration area at NWERC 2024. Photo by Maarten Sijm

The plan is as follows: during the contest registration, where they get their shirts and goodies, they will first wait for the organizer to lean down to grab a shirt. Then, they will quickly lift up one of the k pieces of furniture and move it around. With the limited time frame and the need to do it silently, each team can only manage to move one piece of furniture.

To make sure that they do not get caught, each piece of furniture has to be back at its original place at the end of the registration. From previous visits, the teams know the dimensions of the room and the amount of furniture in it ahead of time, but they do not know the current arrangement of the furniture. To keep things simple, the two teams want to use the same strategy. This means that both teams will make the same move when they encounter the same layout. After deciding on a common strategy and setting off for the contest, they will not be able to communicate. Thus, they plan to come up with a strategy that ensures that, regardless of the arrangement of the furniture, the team that arrives later will undo whatever the team that arrived first did. For some combinations of room size and amount of furniture in the room this is impossible, so pulling off this kind of stunt would be risky.

At least they know for sure they are the only ones pulling this kind of stunt, so no furniture should move around between the two teams' arrivals. Help them find a strategy for their prank!

Input

The input consists of:

- One line with an integer t ($1 \leq t \leq 10\,000$), the number of test cases.
- For each test case, the input consists of:
  - One line with three integers h, w, and k ($1 \leq h, w \leq 8$, $1 \leq k < h \cdot w$), the height and width of the registration room and the number of pieces of furniture in the room.
  - h lines with w characters, each character being either '`.`' or '`#`', the state of the room. A '`.`' represents an empty area, and '`#`' represents a single movable piece of furniture. It is guaranteed that the room contains exactly k pieces of furniture.

  It is guaranteed that there is at least one piece of furniture and some empty area.

This is a multi-pass problem. Your program will be invoked multiple times, possibly more than twice. Your program must act consistently within each invocation, but also across invocations.

For testing purposes, the number and input of subsequent passes will depend on the output of your submission.

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

Output

For each test case, output “risky” if there is no valid strategy the teams could come up with beforehand that works regardless of how the k pieces of furniture end up being placed. Otherwise, specify how a piece of furniture is moved: first output the position of the piece of furniture you move and then output the position where it should be moved to. Both positions must first specify the row r ($1 \leq r \leq h$, counting from the top) and then the column c ($1 \leq c \leq w$, counting from the left).

Sample Case 1

Sample Input (Pass 1)	Sample Output
3 1 4 2 .#.# 4 4 8 ..#. ###. ..## .#.# 1 3 1 .#.	1 4 1 1 4 4 4 1 risky

Sample Input (Pass 2)	Sample Output
2 1 4 2 ##.. 4 4 8 ..#. ###. ..## ##..	1 1 1 4 4 1 4 4

Problem I
Illuminated Stalls
Time limit: 8 seconds

Mulled wine stall at the Leipzig Christmas Market. CC BY-SA 4.0 by Joachim Köhler on Wikimedia Commons

Year after year, you have strolled through the Christmas market, catching up with old friends over mulled wine. Lately the prices have been absurd, and just a few evenings leave you in financial ruin. This year, you decided to turn things around and open a stall selling mulled wine. But competition is fierce, and it turns out that yet another absurdly overpriced stall is far less profitable than you hoped.

To stand out, you created a difficult puzzle. If a customer solves it, they can drink as much mulled wine as they want for free. Naive customers are often willing to pay more than usual.

This puzzle consists of n line-shaped neon light tubes hanging on the wall. They are oriented either horizontally or vertically. No two horizontal lights overlap or touch and no two vertical lights overlap or touch, but a vertical light can intersect or touch a horizontal light. The player is allowed to rotate and/or move *at most* one light tube in any way they like. The goal is to have at least one illuminated neon light square. All four sides of this square have to be fully covered by neon lights, but the lights can be longer than the sides of the square.

Lights are allowed to lie in the interior of the square or intersect its sides. The light that is moved is allowed to be placed such that it touches, partially overlaps, or fully overlaps other collinear lights.

You want to make this puzzle as hard as possible, but if there is no valid solution, you will get into trouble with the German lawmakers, as regulations are very strict. Given a puzzle, find out whether there is a way to make a square by moving and/or rotating at most one light.

Input

The input consists of:

- One line with an integer t ($1 \leq t \leq 20\,000$), the number of test cases.
- For each test case, the input consists of:
  - One line with an integer n ($4 \leq n \leq 2 \cdot 10^5$), the number of neon light tubes.
  - n lines, each with four integers x_1, y_1, x_2, and y_2 ($0 \leq x_1, x_2, y_1, y_2 \leq 10^9$, $x_1 \leq x_2$, $y_1 \leq y_2$, and either $x_1 = x_2$ or $y_1 = y_2$, but not both), where (x_1, y_1) and (x_2, y_2) are the endpoints of a light tube.

The total number of light tubes over all test cases is at most $2 \cdot 10^5$.

All light tubes are either vertical or horizontal. For every test case, the initial configuration has no two overlapping or touching horizontal lights, and also has no two overlapping or touching vertical lights. Note that horizontal lights can intersect and touch vertical lights.

Output

For each test case, output "`yes`" if there is a way to make a square by moving and/or rotating at most one light, otherwise output "`no`".

Sample Input 1	Sample Output 1
5 4 0 0 0 1 0 1 1 1 1 0 1 1 0 0 1 0 4 0 0 0 4 0 4 4 4 4 0 4 4 10 10 10 14 5 0 0 0 4 0 4 4 4 3 1 4 1 4 0 4 4 10 10 10 13 5 0 0 0 4 0 4 4 4 4 0 4 4 3 0 4 0 10 10 10 13 9 1 3 6 3 2 1 2 8 2 7 8 7 4 6 6 6 6 7 6 8 5 5 7 5 6 2 6 4 7 3 11 3 9 1 9 5	yes yes no yes yes

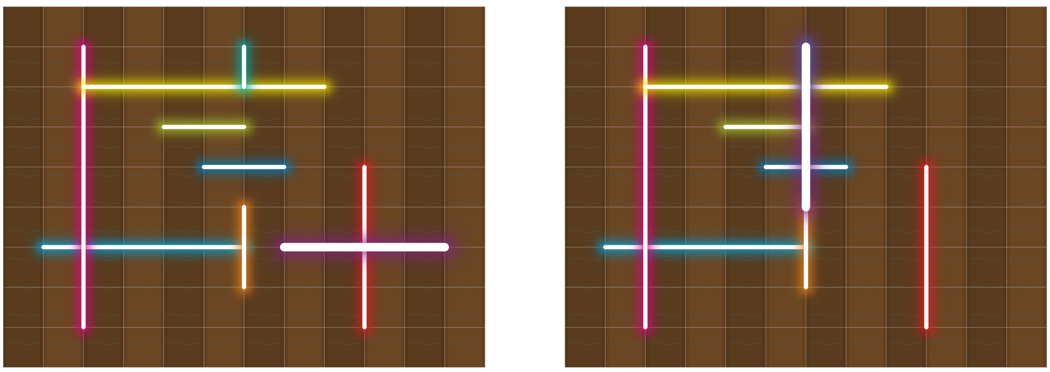


Figure I.1: On the left, the initial configuration of lights of the fifth test case of Sample Input 1 is shown. On the right, the purple neon light was rotated and moved to make a 4×4 square.

Problem J
Juggling Keys
Time limit: 5 seconds

The DOMjudge team working in a shared flat.
Photo provided by the DOMjudge team

Nearly a hundred people are involved in making NWERC possible – organizers, volunteers, jury, the systems and streaming team, and last but not least, the DOMjudge team, who are responsible for the judging system. They come to every NWERC and always have a lot of fun while being there!

The DOMjudge team members like to share a flat during NWERC, but often, there are not enough keys for everybody to have their own. This can make logistics a bit tricky: some people like to head out early for breakfast, others stay out late at the Christmas market, and some may return to the flat for a quick afternoon nap. If someone returns to the flat while another person is already in the flat, they can simply ring the bell and will be let in. However, if someone returns while the flat is empty, they will need to have a key with them.

Given the times when each person is out on a trip, determine when each person should take a key with them so that everyone can get into the flat without being (temporarily) locked out.

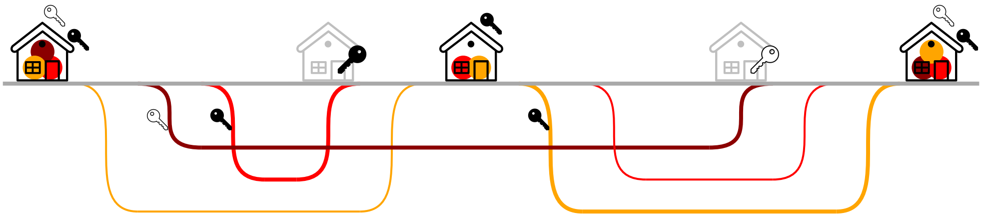


Figure J.1: Illustration of Sample Input 1, with 3 DOMjudge team members, 2 keys, and a total of 5 trips. Trips where the person brings a key are shown in bold. Twice, a person comes home to an empty house and has to use their key to open the door.

Input

The input consists of:

- One line with three integers n, k, and q ($1 \leq k \leq n \leq 10^5$, $1 \leq q \leq 10^5$), the number of DOMjudge team members, the number of keys, and the number of trips.
- q lines, each with three integers p, ℓ, and r ($1 \leq p \leq n$, $0 \leq \ell < r \leq 10^9$), describing a trip where person p leaves at time ℓ and returns at time r.

At any time, at most one person arrives or leaves.

It is guaranteed that for each person, the trips they make do not touch or overlap.

Output

If it is possible to distribute keys such that no one is locked out, output a string of q characters, where each character is either '`0`' or '`1`'. The ith character of the string is '`1`' if the person going on the ith trip (in input order) should take a key with them and '`0`' if they should not take a key with them. Otherwise, output "`impossible`".

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

Sample Input 1	Sample Output 1
3 2 5 3 0 9 1 2 18 2 4 7 3 12 22 2 14 20	01110

Sample Input 2	Sample Output 2
2 1 2 1 2 4 2 1 3	01

Sample Input 3	Sample Output 3
2 1 3 1 1 5 2 2 3 2 4 6	impossible

Problem K
KIT Finding
Time limit: 2 seconds

"*Find the Fox*" is a recent book that contains 200 pages of word search puzzles comprising only the letters '`F`', '`O`' and '`X`'. The special feature of the book is that there is only a single occurrence of the word "`FOX`" throughout all of its pages.

Recall that in a word search the goal is to find hidden words (only one hidden word in this case) in a grid of letters. The words can occur horizontally or vertically or diagonally, as well as forwards or backwards, allowing for a total of 8 different reading directions.

For this year's NWERC, we want to create a baby version of "*Find the Fox*". Your goal in this problem, therefore, is to create a word search grid with given dimensions and containing each of the letters '`K`', '`I`' and '`T`' a given number of times. Similar to the original book, this grid should contain exactly one occurrence of the word "`KIT`".

I	K	I	I	T
K	K	T	K	T
I	T	I	T	I
K	T	T	K	I

Figure K.1: Illustration of Sample Output 1. Flip page for the solution.

Input

The input consists of:

- One line with five integers h, w, k, i, and t ($3 \leq h, w \leq 100$, $k, i, t \geq 1$, $k+i+t = h \cdot w$), where h and w are the height and width the word search grid should have, and k, i, and t specify the required number of occurrences of '`K`', '`I`' and '`T`', respectively.

Output

Output a word search grid according to the given rules. It can be shown that such a word search grid always exists.

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

Sample Input 1

```
4 5 6 7 7
```

Sample Output 1

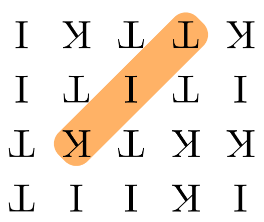
```
IKIIT
KKTKT
ITITI
KTTKI
```

Sample Input 2

```
3 3 1 7 1
```

Sample Output 2

```
III
KIT
III
```



NWERC 2025

Problem L
Last Christmas
Time limit: 1 second

Back cover of Last Christmas, modified. Public Domain by Epic Records / CBS Records Inc. on Wikimedia Commons

A Christmas market is incomplete without live music, and you will find everything from out-of-tune renditions of “Jingle Bells” on a toy xylophone to professional arrangements of Bach’s Christmas Oratorio. This year, Jonathan and Merle also want to earn some money playing Christmas songs, but they cannot even agree on which artists, let alone which songs. Jonathan likes Mariah Carey best, while Merle loves the whole Christmas album by the Goo Goo Dolls.

They decide to settle the discussion using statistics and the following strategy. They take n past Christmas top-10 lists and count how often each artist appears. Note that an artist may appear multiple times on the same list. The artist with the highest number of appearances is clearly the best! If there is a tie, they look at the artist with the highest number of number 1 hits among the artists with the highest number of appearances. If there is still a tie, they look at highest number of number 2 hits among those, and so on.

Who is the best artist for the Christmas season?

Input

The input consists of:

- One line with an integer n ($1 \leq n \leq 100$), the number of top-10 lists.
- n lines, each with 10 strings s ($1 \leq |s| \leq 20$), the first names of the artists of the top-10 list from 1 to 10. Each name only consists of English lowercase letters (`a-z`).[1]

Output

Output the name of the best artist according to the rules described above, or “`tie`” if there is no unique best artist.

Sample Input 1

```
3
elton jose bowie eagles elvis wham nat joni bruce minnie
elton madonna wham elvis andy donny madonna aha chuck judy
madonna heart whitney chuck joni wham john yoko nsync stevie
```

Sample Output 1

```
madonna
```

Madonna and Wham! both appear three times in total, but since Madonna has one more number 1 hit than Wham!, Madonna is the best artist.

[1] If Madonna does not need a last name, why would other artists need one?

Sample Input 2

```
3
maria a b wham c d e f maria wham
wham g h maria i j k l wham m
n o p q r s t u v maria
```

Sample Output 2

```
tie
```

Sample Input 3

```
2
b a b d e a e a b a
b b b e d a d a a a
```

Sample Output 3

```
a
```