Problem A. Sudo

Input file:	`standard input`
Output file:	`standard output`
Time limit:	10 seconds
Memory limit:	256 megabytes

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

Figure 1: An example of a grid of our sudoku variant

You have probably heard of the puzzle game Sudoku, one of the most famous puzzle games in the world, the game is usually present in the puzzle section of most newspapers. The objective is to fill a 9×9 grid with digits so that each column, each row, and each of the nine 3×3 subgrids that compose the grid contain all of the digits from 1 to 9.

Today, we are interested in studying a small variation of the game. We will call this variation Sudo. We want to measure people's performance when the game has different dimensions of the grid and subgrids, and how that would affect the difficulty of the game.

In order to conduct this experiment, we need a Sudo grid generator, as we need to provide all the participants with different grids at different stages, this task would be time consuming if done manually, and for that we need a computer program to do it.

Since you are a computer science expert, can you help us make a Sudo grid generator for our experiment?

Input

The first line contains two integer A, B, the dimensions of the grid. $(1 \leq A, B \leq 20)$

The second line contains two integer N, M, the dimensions of the subgrids. $(1 \leq N, M \leq 20)$

It is always guaranteed that a valid grid exists.

Output

Print an $A \times B$ solved Sudo grid.

Examples

standard input	standard output
6 6 2 3	1 2 3 4 5 6 4 5 6 1 2 3 2 1 4 3 6 5 3 6 5 2 1 4 5 3 1 6 4 2 6 4 2 5 3 1
9 9 3 3	1 2 3 4 5 6 7 8 9 4 5 6 7 8 9 1 2 3 7 8 9 1 2 3 4 5 6 2 1 4 3 6 5 8 9 7 3 6 5 8 9 7 2 1 4 8 9 7 2 1 4 3 6 5 5 3 1 6 4 2 9 7 8 6 4 2 9 7 8 5 3 1 9 7 8 5 3 1 6 4 2

Problem B. Lets Go

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

Go is a strategy board game for two players. It is played with Go pieces (black and white stones) on a Go board that is usually 15x15. Players alternate turns placing a stone of their color on an empty intersection. The goal of the game is to surround more territory than the opponent.

In this problem we consider the go boards to be of given dimensions $N \times M$ (N rows and M columns with the top line being number 1 and leftmost column being 1).

Given the state of a go board after several moves from both players, you are asked to print

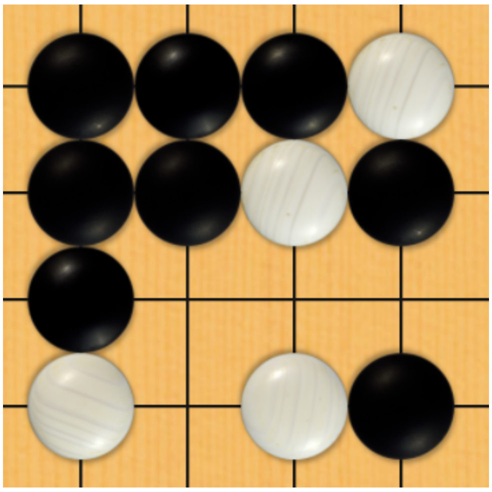
- the size of the maximum territory **white** can capture after placing exactly one **white** go stone on the board.
- the size of the maximum territory **black** can capture after placing exactly one **black** go stone on the board.

A territory is a set of adjacent (in the same line or column and adjacent to each other. diagonal stones are not considered adjacent) go stones of the same colour.

A territory can be captured if and and only if it is completely surrounded by pieces of the other colour or the borders of the board.

In the first sample:

- White can capture 6 pieces if he places a white stone in (3, 2)
- Black can capture one piece by placing a black stone in (3, 3) or (3, 4)



Note: Some stones might already been captured. These should not be included while finding the answer.

Input

The first line of the input contains $1 \leq T \leq 10$ The number test cases.

The first line of each test case consists of two numbers, $3 \leq N \leq 100$ and $3 \leq M \leq 100$, the dimensions of the go board in this test case. N lines follow each consists of a string of size M, the state of the board for this test case.

The strings representing the state of go board will only consist of 3 possible characters

- '.' representing an empty position.
- 'o' if a **white** was placed in this position.
- 'x' if a **black** was placed in this position.

Output

For each test case, print two numbers w b, where w is the the size of the maximum territory **white** can capture after placing exactly one **white** go stone on the board, and b is the size of the maximum territory **black** can capture after placing exactly one **black** go stone on the board.

Example

standard input	standard output
1 4 4 xxxo xxox x... o.ox	6 1

Problem D. The Theory of Minecraft

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

A Minecraft world consists of two parallel worlds, the normal Minecraft world and the nether. Both worlds are represented as 2D grids.

The size of the normal world is 4 times larger than the nether world. More specifically, if the nether world is of size $n \times m$, then the size of the normal world is $2n \times 2m$.

- The block at position $(0 \leq i \leq n-1, 0 \leq j \leq m-1)$ in the nether maps to the position $(2 \times i, 2 \times j)$ in he normal world.
- The block at position (i, j) in the normal world maps to the position $(i/2, j/2)$ in the nether world.

Both worlds can contain the following blocks:

- Start position represented with upper case S, and end position represented with upper case E.
- Land blocks represented with a dot ".".
- Water Blocks represented with an uppercase W.
- Lava blocks represented with an uppercase L.
- Teleports to help you teleport from the nether world to the normal world and vice versa, represented with an uppercase T.

If a teleport is at position (ti, tj) in the nether, then its guaranteed to have a teleport at position $(2 \times ti, 2 \times tj)$ in the real world. Its also guaranteed that there will be no Water in the nether land.

A player starts at the position (sx, sy) in the real world and want to reach the position (ex, ey) in the real world as fast as possible. However the player must respect the following.

- The player can only move in 4 directions: up, down, right, and left
- The player must not go to a Lava block or he dies.
- Going from a land block to any adjacent block takes exactly one unit of time.
- Going from a Water block to any adjacent block takes exactly two units of time.
- Going from a teleport in a world to the corresponding teleport in the other world takes 4 units of time.

Your goal is to tell how fast (how many units of times) the user can go from the start position to the end position, or report that it is impossible to reach the end form the start.

Input

The first line of the input contains two integers $1 \leq n \leq 100$ and $1 \leq m \leq 100$ the size of the grid representing the nether world (thus the real world consists of $2 \times n$ rows and $2 \times m$ cols).

n lines follow, each consists of exactly m characters representing the nether world.

$2 \times n$ lines follow, each consists of exactly $2 \times m$ characters representing the real world.

Example

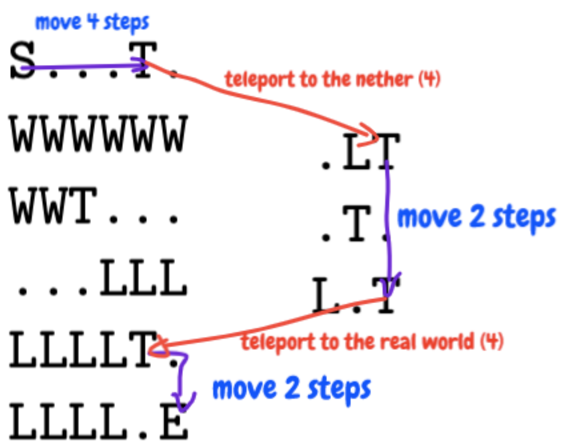
standard input	standard output
3 3 .LT .T. L.T S...T. WWWWWW WWT... ...LLL LLLLT. LLLL.E	16

Note

Description of the sample case:

- The player starts at the position 0 0 in the real world.
- The player moves 4 steps to the position 0 4 in the real world -> 4 units.
- The player uses the teleport at position (0, 4) in the real world to the position (0, 2) in the nether -> 4 units.
- The player moves 2 steps down to position (2, 2) in the nether -> 2 units
- The player teleports from the position (2, 2) in the nether to the position (4, 4) in the real world -> 4 units.
- The player move to the end position (5, 5) in the real world -> 2 units.

The sum of the units used are 4+4+2+4+2 = 16. The player cannot reach the end position without going to the nether as there is Lava in the way. Also if the player try to use the other teleport in the real world the answer will be 17 because of the water surrounding the teleport.



Problem E. Weird Malware Analyzer

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

Mehdi is a malware analyst. He made a weird software to analyze n malicious programs (malware). It accepts only a list of malware with increasing risk level where each malware has a unique risk level from 1 to n. The i^{th} malware has a risk r_i.

In order to make the list compliant to his weird analyzer, he applies the following approach: **while the malware list is not sorted, he removes the malware with the lowest risk level.**

For instance, if he is given 4 malware in this order $[2, 1, 3, 4]$, the list is not in increasing order so he removes the malware with lowest risk level. Thus, the list will become $[2, 3, 4]$ which is in increasing order. So, he needed to remove only one malware to make the list compliant with the weird analyzer's input rules.

Your task is to help him count the number of malware he will need to remove to make it compliant with his weird malware analyzer.

Input

The first line contains a single integer T donating the number of test cases. $(1 \leq T \leq 50)$.

First line of each test case contains a single integer n denoting the number of malware to be analyzed. $(1 \leq n \leq 10^5)$.

Second line of each test case contains n distinct space-separated integers denoting the risks of the malware. $(1 \leq r_i \leq n)$.

Output

For each test case print the number of malware that Mehdi needs to remove from the input list of malware.

Example

standard input	standard output
3 5 1 2 3 4 5 4 2 1 3 4 6 6 1 2 3 4 5	0 1 5

Problem F. Y and Malware Detection

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Y is a security engineer at company X. Company X is specialized in malware analysis and detecting new emerging malware. As you may know, in computer security, many malware are descendants (variants) of other malware that were discovered in the past. Y's boss assigned her/him a task. She is given a list of malware families with links to each other.

Your task is to help Y answer Q queries in the format $query(x, y)$ where x and y are malware IDs specifying the **most precise ID of the malware they are both variants of**.

Input

The first line contains t which is the number of tests cases ($t \leq 50$).

For each test case, there is a number n denoting the number of links between malware ($2 \leq n \leq 10^4$). The following n lines contain two numbers x and y denoting that x is a malware ID of a descendant of y and vice-versa ($x, y \leq 10^4$).

The next line contains a number q denoting the number of queries Y needs to answer ($q \leq 10^4$). The following q lines contain two numbers x' and y' ($x', y' \leq 10^4$).

Output

Print q numbers separated by space for each test case. The number at i-th position is the answer to the i-th query.

Example

standard input	standard output
1 4 1 2 1 3 2 4 2 5 6 4 5 3 4 5 3 2 4 3 2 4 4	0 1 1 0 1 4

Note

All the links are bidirectional.

It is guaranteed that there are no cycles in malware links.

Malware IDs are one-based.

Problem G. Unfinished Artwork

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

Serri is a talented photographer. She is creating a storyboard about women empowerment. She puts all the pictures of her story on a board in order to connect them by lines. However, she gets overwhelmed by the amount of work she has and some links are missing in order to link all the pictures.

Your task is to help her find k the minimum number of links required to connect all the pictures of her storyboard.

Input

First line contains $1 \leq n \leq 10^5$ and $0 \leq m \leq 10^5$, number of pictures and number of existing links respectively. Next m lines contain each two integers (u, v) denoting a link between picture u and picture v.

Output

Print k, the minimum number of links required to connect the pictures.

Examples

standard input	standard output
3 0	2
5 3 2 3 4 3 1 4	1
4 3 1 2 3 2 1 4	0

Note

IDs are 1-based.

Problem H. Tadamoncovid

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

During this pandemic, citizens got messages on their phones to go get their money from ATMs. A security agent in front of an ATM was distributing numbers to organize the waiting line. However, things got messed up, so the line got random organized.

The security agent wants to re-organize the line. Your task is to help him achieve that by counting the number of pairs that should be swapped in order to make the line organized again. **However, the security agent is only allowed to do adjacent swaps**.

A line is said to be organized if the numbers are sorted.

Input

The first line contains an integer N ($1 \leq N \leq 10^5$), the length of the queue. The next line contains N space separated integers ($1 \leq S_i \leq 10^9$) denoting the waiting numbers.

Output

Print the number of described pairs as defined in the problem statement.

Examples

standard input	standard output
3 10 6 4	3
5 2 8 4 20 1	5

Note

In the first sample, for example we can swap 4 with 6, then 4 with 10, and then 6 and 10.

Problem I. Amina and her Patients

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Amina is an amazing medical doctor who hates computer science.

She has a handwritten list of patients who are waiting for their turn. She wants to digitalize her list of patients based on the handwritten list and the following information.

The piece of paper is written as follows: First patients are treated based on the emergency level (the highest emergency level first), in case of a tie, the patients are treated based on waiting number (the small waiting number is first), and if there is still a tie (this hospital's waiting number generation machine gets crazy sometimes), then they are treated in alphabetical order (the lexicographically smallest is higher in the computer list).

Amina hates computer science as you know, so she asked you for help.

Input

The first line contains one interger $1 \leq n \leq 10^5$ denoting the number of patients.

Next n lines contains each: the name of the patient, the emergency level e $1 \leq e \leq 10^7$ and the waiting number w $0 \leq w \leq 10^7$.

Output

Output the names of patients in decreasing order.

Examples

standard input	standard output
2 imane 2 4 khalil 7 14	khalil imane
5 omar 3 1 imane 3 3 khalil 3 5 ahmed 4 6 mehdi 7 2	mehdi ahmed omar imane khalil

Note

All names are distinct. A name contains only lowercase letters and contains at most 15 characters.

Problem J. Busy Drivers

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

You are an application developer at a big transportation company, the drivers are having a strike, and you keep getting customer requests, and you want to convince the minimum number of drivers to go serve those customer queries.

A customer query consist of two map locations and a time, the starting and ending point of the journey and the pickup time.

The distance between two locations in the map is computed using the **Manhattan distance** formula: $|x_0 - x_1| + |y_0 - y_1|$

For simplicity, The drivers speed is constant and it's 1 unit per 1 unit of time.

Can you find the minimum number of Drivers that you need to convince so you can serve those requests ?

Input

First line of input is integer $1 \le N \le 999$ nunber of requests.

Next N lines each of them describing a single customer request of the form fx_i fy_i tx_i ty_i $start$, $(fx_i, fy_i) \neq (tx_i, ty_i)$ where fx fy and tx ty are the starting and ending points on the map, and $start$ is the pickup time. $1 \le fx_i, fy_i, tx_i, ty_i, start \le 10^9$

Output

One single line containing the minimum number of drivers needed to serve all the requests.

Examples

standard input	standard output
3 1 2 2 3 1 2 3 4 5 5 1 1 1 2 1	2
3 1 2 2 3 1 2 3 4 5 5 1 1 1 2 100	1

Note

1- You can assume that the driver will be on time for his first request.

2- If the driver comes early before the supposed starting time of the request, he should wait until the specified time then start processing the request. i.e **a request cannot be processed earlier than it's specified time**.