List of Problems

Problem	Title
A	Bring Me The Problems
B	Prisoners
C	Where's My Water?
D	Game …
E	Antidote
F	Accident
G	UAV
H	Ceiling
I	Morphology
J	Sara
K	Dr. Hamid-Zadeh
L	Lazy Sorting

A: Bring Me The Problems

Recently, Bring Me The Problems group are planning to release their new studio album. This year they acquired a group of people and asked them to listen to all of the new tracks from their upcoming album. They are planning to publish only one single track that most people of the group like as a new single of the album (for promotional purposes). The problem is most artists can only sing and are very bad at math so as a professional programmer you need to help them find their most liked track to release as a new single.

Input

The first line of tests contains an integer t ($t \leq 100$), number of test cases. The first line of each test case contains an integer n ($1 \leq n \leq 100$), the number of tracks then follows n lines each includes a string s and an integer l, the name of track and number of likes it gets ($1 \leq |s| \leq 10, 0 \leq l \leq 10{,}000$). Note that the name of each track contains only lowercase English letters and no track got the same amount of likes.

Output

For each test, print the most liked track in a separate line.

Sample Input	Sample Output
2 2 ouch 5 medicine 10 3 wonderful 32 life 21 mantra 43	medicine mantra

B: Prisoners

Alice* and Bob are arrested for some crime and are thrown in two different cells. They want to develop an escape plan, but unfortunately all communications between them are arbitrated by a warden named Wendy. She will not let them communicate through encryption and if she notices any suspicious communication, she will place them in solitary confinement and thus suppress the exchange of all messages. Alice and Bob run a key-exchange protocol as follows and use the generated keys to hide meaningful information in some harmless cover.

Alice makes a normal string, S_A, and sends it to Bob. Bob does the same and sends back S_B. Consider s_a and s_b as longest two substrings of S_A and S_B, respectively, with same length that share exactly same characters. Substring is a consecutive sequence of characters from a string. Alice uses s_b to embed her message in a normal cover and Bob uses s_a.

Your task is to tell us the length of the keys they will use.

Input:

The first of input contains number of test cases, t ($t \leq 100$).

Each input consists of two lines which contain S_A and S_B ($1 \leq |S_A|$ and $|S_B| \leq 4000$) of solely English lowercase characters.

Output:

Output a single integer in one separate line as the length of shared keys. A zero should be printed out in case of no key can be generated and Alice and Bob should restart the protocol.

Sample Input	Sample Output
2 restful fluster rxtgleuncanxrt omeclangun	7 8

* The story is partly from "Information hiding for Steganography and digital Watermarking" book, S. Katzenbeisser et al.

C: Where's My Water?

Swampy is an alligator lives in a city sewer system. He loves to be clean, unlike most of the alligators, but Cranky and the other alligators have now sabotaged and broken Swampy's water supply.

The water supply is a rectangle of w width and h height. The water flow starts at north side of the water supply and Swampy is at the south side. Cranky tries to block the water flow by building several walls inside the water supply. Each wall is built as a straight line between two points. Multiple walls can cross one another and also no wall touches the north and south of the water supply. In this problem, you should help Swampy to destroy Cranky walls and take a shower. Swampy can take shower if water reaches any point of south side.

For this purpose, Swampy gave you a hammer that you can destroy at most one point of wall (or walls in case of crossing). If any wall crosses that point they will get destroyed as well. The water is a liquid and can pass the walls if there's even a small hole in it. Given the coordinates of the walls, you should tell Swampy that you can provide water for him or not.

Input

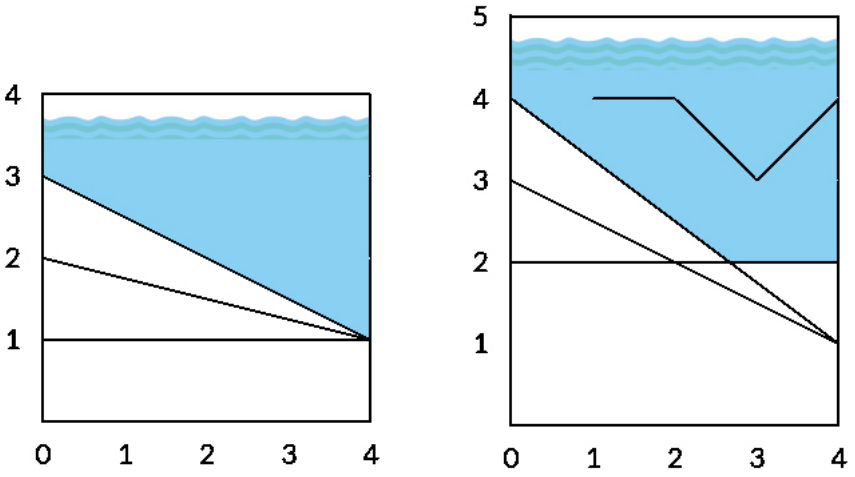
The first line of input contains an integer t, number of test cases, ($t \leq 50$).

The first line of each test contains three integers w and h, width and height of the water supply ($1 \leq w, h \leq 1000$) and n, the number of walls that Cranky has built ($1 \leq n \leq 500$). Then n lines follow each of them contains four integers x_1, y_1, x_2 and y_2 ($0 \leq x_1, x_2 \leq w$ and $0 \leq y_1, y_2 \leq h$) the starting point and finishing point coordinates of each wall that Cranky has built.

Output

For each test print in a separate line "Yeah" (without quotation marks) if you can provide water to Swampy and "Sorry" (without quotation marks) if you can't.

Sample Input	Sample Output
2 4 4 3 0 1 4 1 0 2 4 1 0 3 4 1 4 5 6 0 3 4 1 0 2 4 2 0 4 4 1 4 4 3 3 2 4 3 3 2 4 1 4	Yeah Sorry



Water supply representation of the two sample test cases of Problem C

D: Game …

Let's play the following two player game ☺. Based on throwing a dice one player starts the game. Starter comes up with 6-digits numbers and the follower player finds the closest palindromic number (reading the same backwards and forwards) to it. Each time the follower successfully finds the number, he gains +1 and if he fails -1. By the first failure starter and follower change their roll.

Input

First line of input contains the number $t \leq 5000$ as the number 6-digits numbers and following by t lines each of which contains a 6-digits number with first digit ≥ 1.

Output

Corresponding to each input number, output closest (in terms of absolute difference) 6-digits palindrome number in a separate line (first digit ≥ 1). If there are two numbers with equal absolute difference, output the smaller one.

Sample Input	Sample Output
3 480852 467331 617032	481184 467764 616616

E: Antidote

Wait! Did you feel sick? That's because viruses are planning to attack your body! But don't worry because white blood cells are coming to defend against the viruses. "The War Is On" in your body and you can feel a little of the heat! "Come on boys let's connect!", the leader of white blood cells said. Each white blood cell connects to some other white blood cells. They can connect to each other with a magic cord that horrifies the viruses!

Viruses planned to destroy the most important white blood cell which is the one whose removal disconnects the most pairs of white blood cells. After a while, they did this. The most important white cell is gone but don't lose your hope! The antidote is coming! Antidote does a simple job which connects two white blood cells in a way that the number of disconnected white blood cell pairs is as small as possible.

For this problem, you should count the number of days you are sick before using antidote and after using the antidote. The number of days you are sick without antidote is the number of pairs of white blood cells that are disconnect after the viruses attack and the number of days you are sick with the antidote is the number of pairs of white blood cells that are disconnect after you use an antidote.

Input

The first line of input contains an integer t, number of test cases ($t \leq 50$). The first line of each test contains an integer n ($2 \leq n \leq 10{,}000$), the number of magic cords, then follows n lines each containing two integers in the form of $x\ y$, which means white blood cells x and y are connected ($0 \leq x, y \leq 10{,}000$). All connections are supposed two-way, no connection appears more than once and there is no loop in connections between white blood cells.

Output

For each test print two numbers, the number of days you are sick without using antidote and the number of days you are sick with help of an antidote. It is guaranteed that there is exactly one possible solution for each test case.

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

F: Accident

Assume a motorbike race is holding at a wide highway. Motorbikes travel from north to south in straight vertical lines. All of a sudden, a snake unconscious of the event wants to pass the highway from west to east (perpendicular to bikes line). Assume the highway as a 2D matrix with origin cell (0,0) at top-left corner and cells positive order from left to right and top to bottom. Each bike travels in one fixed column of the matrix and the snake keeps its row all over its journey. All bikes travel v cells towards south after the snake moves one cell ahead to the right. Each bike occupies one cell and the snake occupies w cells. As soon as a bike passes the end of its line in the south another one from the same team starts the exact line from the north. Therefore, there is exactly one bike in the line in all times. Please let us know whether there is a way that the snake can pass the highway safe or not. At time 0 the snake is completely out of highway and it may enter the highway from any arbitrary row.

Input

The first line of input contains an integer t, number of test cases ($t \leq 50$). Each test case starts with a line containing four integers b, ℓ, v and w where b ($1 \leq b \leq 2 \times 10^5$) is the number of motorbikes, ℓ ($1 \leq \ell \leq 2 \times 10^5$) is the length of highway in terms of matrix cells (the highway width is assumed fixed at 2×10^5), v ($1 \leq v \leq 500$) is motorbikes speed in terms of number of cells in each move and w ($1 \leq w \leq 500$) is the length of snake.

Each of the next b lines contain two integers describing one motorbike where x and y ($0 \leq x < 2 \times 10^5$ and $0 \leq y < \ell$) are the column of the bike and its initial row when the snake enters the road.

Output

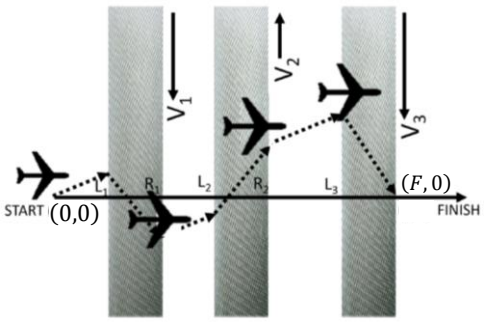
If it is possible for the snake to pass the race route print "safe" (without quotation marks) in one separate line. Otherwise, print "accident" (without quotation marks).

Sample Input	Sample Output
2 2 10 2 3 3 4 5 8 1 5 2 2 1 1	safe accident

G: UAV

An unmanned aerial vehicle (UAV), commonly known as a drone, is an aircraft without a human pilot aboard [Wikipedia]. Consider a special UAV that only flies straight at a constant speed v and also consider a special 2D plane that our special UAV wants to reach to $(F, 0)$ starting from $(0,0)$. Along its route through the special 2D plane, the UAV faces n wind tunnels, inside each of them wind is blowing from north or south (aligned with y-axis) and affects UAV's position while passing.

To get the UAV to go from start to finish, you need to adjust its orientation at start. Assume you can choose this direction correctly and the UAV arrives exactly at destination, how long does it take to get there? If you think the tunnels are slowing down the UAV too long (longer than $2F/v$), please keep us informed.



Input

The first line of input contains an integer t, number of test cases ($t \leq 20$). Each test case contains three numbers; n, F and v ($0 \leq n \leq 100; 1 \leq F \leq 1{,}000{,}000; 1.0 \leq v \leq 100.0$). The next n lines explain the wind tunnels and contain three numbers in each line; s_i, e_i and v_i ($0 \leq s_1 < e_1 \leq s_2 < e_2 \leq \cdots \leq s_n < e_n \leq F; -100.0 \leq v_i \leq 100.0$) stand for start and end of i-th tunnel on x-axis and wind speed inside it. A positive number means wind toward north and a negative number means wind toward south.

Output

Output on a single line the travel time of UAV (in exactly three decimal places). If the UAV is late to destination output "late" (without quotation marks).

Sample Input	Sample Output
3 1 806873 66 91411 631975 -57.5 4 403864 48 58621 58692 -9.5 146285 175147 -76.6 194288 205342 -70.8 244093 343424 -57.3 1 670764 22.4 113447 642610 -64.8	15055.988 9411.393 late

H: Ceiling

Omid is a tile installer. Last week he was proposed a project to tile ceilings of corridors of a big building with 40 x 40 cm^2 and 40 x 80 cm^2 tiles. All corridors have fixed 120 cm width but different lengths. On the ceiling, there are some points the electric wires come out for lights and therefore Omid should not put any tile there. The tiles should be always aligned along the two sides of corridor, they won't extend the corridor and always the edges of tiles are 40 cm multiples away from any edge of corridor. He needs some help to see for each corridor with a specific length, in how many different ways he can tile.

Input

The input starts by an integer t in a separate line indicating the number of test cases ($t \leq 30$).

For each test case, in the first line two integers ℓ and n indicate the length of corridor in centimeters ($40 \leq \ell \leq 960$) and number of lamps we have in the corridor. Next line contains n pairs of coordinates $x\ y$ for lamp locations $0 < x < \ell$ and $0 < y < 120$. Lamps are not located at integer coordinates and coordinates are rounded to two decimal places. If $n = 0$, the test case has only one line.

Output

For each test case print the number of ways Omid can tile the ceiling in a separate line.

Sample Input	Sample Output
2 120 1 9.6 28.4 840 0	67 25080160016800177

I: Morphology

A binary image is a digital image that has only two possible values for each pixel. Typically, the two colors used for a binary image are black and white. Binary morphological image processing is the processing of binary images consists of a set of operators that transform image according to size, shape, convexity, connectivity, and etc. characteristics. One of these morphological operators is boundary extraction. The goal of boundary extraction is to find the pixels that are on the boundary of objects in the image.

Suppose we have a set of $m \times n$ binary images that after applying a special boundary extraction algorithm, we get closed contours without any intersection with each other and each pixel on contour has at most two 8-adjustent pixels (8-adjacency means two pixels are adjacent if they share either an edge or a corner). Your task is to count the number of contours.

Input

The first line of input indicates t as the total number of test cases ($t \leq 30$).

For each test case, the first line is space-separated m and n ($1 \leq m, n \leq 100$). This follows by m lines each of them contains n characters. A w denotes a white pixel, a b denotes a black pixel. Pixels on contours are white and have exactly two adjacent white pixel.

Output

A single integer in a separate line representing the number of closed contours in the input image.

Sample Input	Sample Output
1 6 6 bwwwwb wbbbbw wbwwbw wbbwbw wbbbbw bwwwwb	2

J: Sara

Sara is 5 years old and she loves solving Sudoku. As you for sure know, Sudoku is a logic-based, combinatorial number-placement puzzle. 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 (also called "boxes") contains all of the digits from 1 to 9. The puzzle setter provides a partially completed grid, which for a well-posed puzzle has a single solution.

2	9	5	7	4	3	8	6	1
4	3	1	8	6	5	9	2	7
8	7	6	1	9	2	5	4	3
3	8	7	4	5	9	2	1	6
6	1	2	3	8	7	4	9	5
5	4	9	2	1	6	7	3	8
7	6	3	5	3	4	1	8	9
9	2	8	6	7	1	3	5	4
1	5	4	9	3	8	6	7	2

Sara's father wants to teach her addition and he wants to encourage Sara to practice adding numbers by changing her favorite game a little bit. He divides the Sudoku into regions and adds a constraints to the game that all numbers in one region have to sum up to a special number.

Sara's father is not very good in Sudoku and he may define regions and their sum in a way that the puzzle becomes insoluble ☹. Your team should help Sara's father to check whether a completed grid meets all constraints of row, column, box and region.

Input

The input starts by a number t indicating the number of test cases ($t \leq 120$).

Then for each test case the 9×9 grid of special Sudoku is presented. Each cell of the grid including its borders is 3×5 (the neighbour cells share the borders). First and last rows of each cell consists of: a #, a space, a #, a space and then another #. Second row is presented as: a #, a space, the number (1 to 9), another space and another #. Regions are defined by replacing the middle # with a space of the shared border between each of two cells of the region.

Following the grid, in each line one region is defined by three numbers, r, c and s indicating row and column of one cell of the region and the region corresponding sum. Note that the sum value is presented for only one randomly selected cells of the region.

Output

For each test case, print in separate line the test case number and then following a space if the Sudoku meets all row, column, box and region constraints, print "true" (without quotation marks) and otherwise print "false" (without quotation marks).

Sample Input

```
1
# # # # # # # # # # # # # # # # # # #
# 2   9 # 5   6 # 4 # 8   7   1 # 3 #
#   #   # # # # # #   #   #   # # #   #
# 3   8   7   5   1 # 2   4 # 9   6 #
# # # # # # # # # # # # # # #   #   #
# 4 # 6   1   9   7   3 # 2   8   5 #
#   # # #   # # # # # # # # # # # # # #
# 5   2 # 4 # 8   6 # 7   9   3   1 #
#   #   # # #   #   #   #   #   # # #
# 1   3   6 # 4   9 # 5   8   2 # 7 #
#   #   #   #   #   # # # # # # # # #
# 8   7   9 # 3   2   1   5 # 6   4 #
# # # # # # # # # # # # # # #   #   #
# 7   4 # 8   2   3   6   1 # 5   9 #
#   #   # # # # # # # # # # # # # # #
# 6   1   2 # 7   5   9   3 # 4 # 8 #
# # # # # # # # #   #   #   # # #   #
# 9   5   3   1 # 8   4   6 # 7   2 #
# # # # # # # # # # # # # # # # # # #
9 4 18
1 4 11
3 3 30
8 3 20
6 5 38
9 6 42
8 9 41
2 6 22
8 8 4
5 7 35
7 5 20
3 7 33
4 2 45
2 5 39
5 9 7
```

Sample Output

```
1 true
```

K: Dr. Hamid-Zadeh

Dr. Hamid-Zadeh is head of computer engineering and information technology department at Sadjad University of Technology. He teaches algorithms design course and in his course he uses a new way of scoring. He has three main grades: A, B, and C. Obviously, A is highest and C is the lowest. But he also scores in more details, like grade A itself can have three subgrades AA, BA and CA where AA is higher than CA. However all A sub-grades are higher than B grades. Still within subgrades there can be more detailed scores like CBABA. Your team is supposed to help Dr. Hamid-Zadeh to sort the students of his algorithms design class in this scoring system.

Note that when you reach the lowest level of detail, you should assume that further subgrades are equivalent to B subgrade of the previous level of detail. Therefore, A and BA are equivalent and so are BCB and CB.

Input:

The first line of input contains t, number of test cases ($t \leq 20$).

Each test on its first line contains p ($1 \leq p \leq 1{,}000$) as the number of student names to follow. Each of the following p lines contains the name of a student (solely a sequence of lowercase letters), space, and then the grade of the student. There are not two students with same name and input lines are not longer than 256 characters.

Output:

For each test case print in one separate line test case number following the name of p students in separate lines in decreasing order of their grades. If two student have same grade, they should be printed out in lexicographic order by their name.

Sample Input	Sample Output
1 5 maryam AACB davood BBCB qazal AA changiz CC unes BCB	1 qazal maryam davood unes changiz

L: Lazy Sorting

Consider an input array of n integers that contains a permutation of $\{1,2,..,n\}$. Having a lazy sorting algorithm that can only sort some subarrays, we are interested to know if we can sort every permutation correctly. A subarray is a contiguous portion of the original array, which can be specified by a start and end index.

Input

The first line contains the number of test cases t ($t \leq 100$).

For each test case, first line contains two space-separated integers n ($1 \leq n \leq 50$) and m ($1 \leq m \leq 50$) as the size of input array and number of subarrays that our algorithm can sort, respectively.

The following m lines of input will contain two integers s_i and e_i ($s_i \leq e_i$) describing the start and end indices of the subarray that lazy algorithm will sort.

Output

For each test case, if every permutation is correctly sorted, print the string "correct" (without quotes) in a separate line. Otherwise, print n space-separated integers, representing the lexicographically smallest permutation of $\{1,2,..,n\}$ that can't be sorted.

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