Problem A. Bags of Candies

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

There are people that won't be happy with solving a problem, unless the actual task is obscured by an elaborate and somewhat unnecessary story. If you are one of these people, then this problem is for you.

Your life as an Environment Specialist at a local candy factory couldn't be more sweet, both figuratively, and literally. And it's bound to get even sweeter, as you were promised a promotion if you can halve the plastic footprint of the company – which of course you're planning on doing.

Right now, the factory uses enormous amounts of plastic packaging – every day, it produces n flavors of candy, numbered from 1 through n, and they wrap each flavor in a separate bag. Your brilliant idea is to simply start packing two flavors of candy per bag, and hence halve the amount of bags being used!

Not so fast, though – it's not that simple. The i-th flavor is always produced in the quantity of i candies, now packed into one bag. If two flavors, say i-th and j-th, are packed into a single bag, you need to ensure that the candies in the shared bag can be evenly and fairly divided between a group of friends of some size. Formally, you want i and j to have a common divisor greater than 1.

You can pair some flavors of candies to be together in one bag, and place the remaining flavors in their individual bags. What is the least number of bags that you will need?

Input

The first line of input contains the number of test cases z ($1 \le z \le 5$). The descriptions of the test cases follow.

The only line of each test case contains a single number n ($2 \le n \le 10^{11}$) – the number of flavors of candies produced at the factory.

Output

For each test case, output a single line, containing the least number of bags needed.

Example

standard input	standard output
2 4 9	3 6

Problem B. Binomial

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

There are people that wont be happy with solving a problem, unless the actual task is obscured by an elaborate and somewhat unnecessary story. If you are one of these people, then this problem is NOT for you.

You are given a sequence of non-negative integers $a_1, a_2, \ldots, a_n$. You need to find the number of pairs (i, j) such that $1 \le i, j \le n$ and $\binom{a_i}{a_j}$ is odd.

Note that $\binom{n}{k}$ is the number of ways, disregarding order, that k objects can be chosen from among n objects. In particular, if $n < k$ then $\binom{n}{k} = 0$.

Input

The first line of input contains the number of test cases z $(1 \le z \le 10)$. The descriptions of the test cases follow.

The first line of each test case contains the number of elements in the sequence n $(1 \le n \le 10^6)$.

The second line contains n integers a_i $(1 \le a_i \le 10^6)$ – the elements of the sequence.

Output

For each test case, output a single line which contains a single integer – the answer for the problem.

Example

standard input	standard output

2
3
1 5 6
3
1 1 1
	4
	9

Problem C. Bookface

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Taking on an ambitious project? Check. Working tirelessly, day and night, to deliver it before the deadline? Check. Committing your code every day in small chunks? Check. And after all that work, you, the most ambitious software developer at Bookface to date, thought that there was nothing standing between you and your promotion. You couldn't have been more wrong.

Before sending your promotion package to the management, you decided to ask your coworker Little Franiu to take a look. And one quick look was enough to see the problem. "At Bookface, you have to move fast" – said Franiu – "move fast and change things. You can't be predictable and ship similarly-sized commits every day like that!".

You looked at the number of lines you added on each of the n days of the project, and realized Franiu had a point. If we denote this count for the i-th day by c_i, then all the c_i values turned out to be relatively close to each other. Fortunately, your friend also had an idea how to fix this – you can simply rewrite your commit history to make it look better!

Together with Franiu, you chose a value of d, and decided that you want $|c_i - c_j| \geq d$ to hold for any $1 \leq i < j \leq n$. To that end, you can select a single day, and either add or remove one line of code in the commit made on that day. You can perform arbitrarily many such operations, and each takes you 1 second. How much time do you need to accomplish your goal? Of course, the number of lines of code in a commit has to always stay non-negative.

Input

The first line of input contains the number of test cases z ($1 \leq z \leq 100\,000$). The descriptions of the test cases follow.

The first line of each test case contains the number of days of the project n ($1 \leq n \leq 200\,000$) and the chosen constant d ($1 \leq d \leq 10^6$). The second line contains n numbers c_i ($0 \leq c_i \leq 3 \cdot 10^{11}$) – the number of lines of code added on the i-th day.

The total number of days in all test cases does not exceed 10^6.

Output

For each test case, output a single line containing a single integer – the number of seconds needed to accomplish your goal.

Example

standard input	standard output

2
4 1
0 0 0 0
4 10
1 100 5 10
	6
	16

Problem D. Clique

Input file:	*standard input*
Output file:	*standard output*
Time limit:	25 seconds
Memory limit:	512 mebibytes

Consider a circle divided into 10^6 arcs numbered clockwise from 1 to 10^6. Moreover, there are n segments on the circle, each spanning a connected interval of arcs.

Find the size of the largest set of segments such that every two share at least one common arc.

Input

The first line of input contains the number of test cases z. The descriptions of the test cases follow.

The first line of each test case contains the number of segments n ($1 \le n \le 3000$). The following n lines contain two integers l_i and r_i ($1 \le l_i, r_i \le 10^6$) – the first and last arc of the i-th segment if we traverse the circle clockwise. No segment contains the entire circle, and no two segments coincide.

The sum of n over all test cases does not exceed 24000.

Output

For each test case, output a single line containing a single integer – the size of the largest set of segments such that every two of them intersect.

Example

standard input	standard output

1
4
1 4
4 5
5 2
3 3
	3

Problem E. Contamination

Input file:	*standard input*
Output file:	*standard output*
Time limit:	10 seconds
Memory limit:	512 mebibytes

The Third World War broke out! In a short period of time there were nuclear explosions all around the world, and the explosions contaminated large areas of Earth. Radioactive contamination wiped out large populations of animals, some species became extinct, and some are now endangered.

For a number of endangered species, a pair of animals of that species was observed. We would like to know whether these two animals can safely meet. What makes the answer to that question harder is the fact that each species can only survive in a specific range of latitudes.

We model the world as a plane with standard Cartesian coordinates. The area of contamination of each of the n nuclear explosions is a circle centered at the point (c_x, c_y) with the radius r. No species can survive being at distance r or smaller from the point (c_x, c_y). The areas of contamination of the explosions are pairwise disjoint (It would be a waste to contaminate one place in the world twice!).

You are given q queries of the form $(p_x, p_y, q_x, q_y, y_{min}, y_{max})$, where (p_x, p_y) and (q_x, q_y) are the coordinates of two animals of the same species, and y_{min} and y_{max} define the range of latitudes in which the species can survive. You have to determine whether it is possible to get from the point (p_x, p_y) to the point (q_x, q_y), avoiding the areas of contamination and traveling only through points (x, y) with $y_{min} \le y \le y_{max}$. You may assume that the points (p_x, p_y) and (q_x, q_y) are not in any area of contamination.

Input

The first line of input contains two integers n, q $(1 \le n, q \le 10^6)$ — the number of explosions and the number of queries.

The next n lines contain the descriptions of the explosions. Each description consists of three integers c_x, c_y, r $(-10^9 \le c_x, c_y \le 10^9,\ 1 \le r \le 10^9)$ — the coordinates of the center and the radius of the area contaminated by the explosion. Remember that these areas are pairwise disjoint.

The following q lines contain the queries about animals. Each query consists of six integers $p_x, p_y, q_x, q_y, y_{min}, y_{max}$ $(-10^9 \le p_x, p_y, q_x, q_y, y_{min}, y_{max} \le 10^9,\ y_{min} \le p_y, q_y \le y_{max})$, with the meaning explained above.

Output

For each query, output in a separate line "YES" if the pair of animals can safely meet, or "NO" otherwise.

Example

standard input	standard output

3 3
3 3 2
7 7 3
12 5 2
1 4 14 4 2 6
1 4 14 4 4 7
1 4 14 4 3 9
	YES
	NO
	YES

Problem F. The Halfwitters

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

You recently did a fabulous prank on your superior officer. And it worked extremely well – you were promptly demoted, relieved of your post and assigned to command the elite platoon of carefully hand-picked soldiers, the famous Halfwitters. It is said that no Halfwitter has ever been suborned by the enemy or disgraced themselves in battle – they simply do not understand the concepts of retreat or betrayal. And no Halfwitter has ever served in a place where outside temperature could fall below their IQ.

The platoon, consisting of n soldiers, is standing in a row before you. You would like the men to stand from the tallest one (number 1) to the smallest (number n). This, however, has yet to be explained to the platoon. Right now, they are standing in their favourite order of whoever-was-the-first-to-finish-their-dessert. You may take the following three actions:

- Tell any two neighbouring soldiers to swap their places. This takes exactly a minutes of explaining.
- Command the entire platoon to reverse the order of the row – it is a hard maneuver, but they have already drilled it, and need b minutes of reminding.
- Lose your temper and shout for c minutes. This creates a lot of panic and confusion, and after the shouting stops, the soldiers rearrange themselves into a completely random order.

Assuming the best strategy possible, what is the expected time to achieve the desired order of $(1, 2, \ldots, n)$? You will drill the platoon for d days, every day starting with possibly different order, because of various available desserts. Compute the answer for each day.

Input

The first line of input contains the number of test cases z. The descriptions of the test cases follow.

Every test case starts with a line consisting of five integers n, a, b, c, d ($2 \leq n \leq 16$, $1 \leq a, b, c \leq 1000$, $1 \leq d \leq 10\,000$) – the number of soldiers, the costs of each action, and the number of days. Each of the next d lines contains a permutation of the sequence $(1, 2, \ldots, n)$ – the initial order of soldiers on consecutive days. The total number of days in all test cases does not exceed $100\,000$.

Output

For each test case output d lines – for every day, output the expected time needed to arrange the soldiers. As this is a rational number, express it as an irreducible fraction of the form p/q.

Example

standard input	standard output

1
6 1 1 1 3
1 2 3 4 5 6
5 4 3 2 1 6
6 4 2 1 3 5
	0/1
	6/1
	2771/428

Problem G. Invited Speakers

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

At a big conference, there are n invited speakers – leading researchers in the area. Unfortunately, they hate each other wholeheartedly and passionately. None of them would bear to speak later than anyone else – so they all must have lectures at the same time, right after the welcome party. You managed to arrange n different rooms for them to speak, and n tables at the party. The conference site is very big, so we can treat it as an Euclidean plane, and all the tables and rooms can be considered points on this plane.

All the rooms, as well as all the tables are, of course, identical, so you can assign any room or table to any of the scientists. The only problem remaining is how do they get from one of these spots to the other – if any two speakers cross paths, a serious scandal is guaranteed. You can give them very detailed instructions on what path they should take, but they seem to only travel along straight lines. Therefore you must connect n tables with n speaking rooms with polygonal chains (connected series of line segments) such that no two of them cross each other.

Input

The first line of input contains the number of test cases z ($1 \le z \le 200$). The descriptions of the test cases follow.

The first line of each test case contains a single integer n ($1 \le n \le 6$) – the number of invited speakers. The next $2n$ lines contain two integers x_i, y_i each ($|x_i|, |y_i| \le 100$) – coordinates of the i-th spot. The first n spots are tables, and the remaining ones are lecture rooms.

You may assume that all x_i are distinct, all y_i are distinct, and no three points are collinear.

Output

For each test case, output n polygonal chains in the following format: for each chain, output first the number k of its vertices. Then, in k following lines, output the coordinates of consecutive vertices of this chain. The ends of each chain must be two input points: a table and a lecture room. No chain may self-cross, and no two chains can have common points (in particular, every input point must appear at the end of some chain). Also, k should not exceed 100 and the coordinates of all points should be between -1000 and 1000.

Example

standard input	standard output

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

Problem H. Lighthouses

Input file:	*standard input*
Output file:	*standard output*
Time limit:	15 seconds
Memory limit:	512 mebibytes

Lighthouse Island is an island whose shape is a convex polygon. On each vertex of the polygon there is a lighthouse, which provides a beautiful wide sea view. Some pairs of lighthouses are connected by straight tram lines. Some of these lines may intersect, but since there are no switches, a tram which enters a tram line can only exit it by driving to the other end.

Daredevil Vladik recently acquired a tram (through completely non-violent means, description of which is beyond the scope of this problem statement), brought it to the island, and he is planning a ride of his life. He wants to start next to one of the lighthouses, visit several lighthouses in some order by traveling along the tram lines, and then depart the island. Moreover, following Vladik's tracks, the FBI also came to the island. They are keen to meet Vladik – mostly to say hi, and talk about how operating a tram without a valid tram driving license is a completely sensible thing to do. Since Vladik would like to avoid meeting FBI for now, whenever he visits some point on the island, he shall never come to the same point again. In other words, Vladik's trip should form a polygonal chain without self-intersections.

Vladik wants the route of his trip to be as long as possible (in terms of total euclidean length), so he can brag to other Vladiks about his stunt. Help him – given the map of the island, find the longest possible trip he could take without visiting any point on the island twice.

Input

The first line of input contains the number of test cases z. The descriptions of the test cases follow.

The first line of every test case consists of an integer n ($3 \le n \le 300$) – the number of lighthouses. Then, n lines follow, each one containing two integers: x_i and y_i ($-10^9 \le x_i, y_i \le 10^9$) – the coordinates of the i-th lighthouse. The lighthouses are listed in the same order as they appear when moving along the borderline of the island counterclockwise. No three lighthouses lie on a common line. The next line contains an integer m ($0 \le m \le n(n-1)/2$) – the number of tram lines. Then, m lines follow, each one containing two integers: v_i and u_i ($1 \le v_i \ne u_i \le n$) – the indices of lighthouses connected with a tram line. Every undirected pair of lighthouses is connected by at most one line.

The total number of lighthouses in all test cases does not exceed 3 000.

Output

For each test case output one number: the maximum length of Vladik's trip. Your answer will be considered correct if the absolute or relative error does not exceed 10^{-6}.

Example

standard input	standard output

2
4
0 0
1 0
1 1
0 1
3
1 3
2 4
3 4
4
0 0
1 0
1 1
0 1
4
1 4
4 3
3 2
2 1
	2.414213562373
	3.000000000000

Problem I. Sum of Palindromes

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

For a 7-years-old you, a day would not be complete without a fight with your little sister. Today's fight is about palindromes: she has recently become very passionate about the topic, while you think this is an exceptionally stupid choice of topic to be passionate about. Obviously, you couldn't have missed a chance to express this opinion out loud.

"Palindromes are awesome, you ignorant", she shouts, "for example you can add palindromes and the result will always remain a palindrome! Like $242 + 515 = 767$! Or $111 + 222 + 333 = 666$. See, stupid?"

There it is! You could easily end this pointless conversation by showing her some palindromes which sum to a non-palindrome, but it would be too simple, too easy... You don't want to win with her, you want to *humiliate* her. So instead you ask her to produce a number, *any* number, and you'd show her that it is actually a sum of just a few palindromes. Let's say, no more than 25 palindromes: you would absolutely hate it if you needed to add up so many palindromes that she lost interest before you finished the summation.

Clearly, you've underestimated your enemy. You sit by your desk and you look at your number... 100 000 digits long...

Input

The first line of input contains the number of test cases z ($1 \leq z \leq 20\,000$). The descriptions of the test cases follow.

A test case consists of a single line, containing a positive integer with at most 100 000 decimal digits.

The total number of digits in all numbers does not exceed 3 000 000.

Output

For each test case, first output a line containing the number of palindromes k, which should be between 1 and 25 inclusive. Next, k lines should follow, each containing a positive integer being also a palindrome. Leading zeroes are not allowed (and thus, for example, 3630 **is not** a palindrome). The sum of all the numbers should equal the input integer.

Example

standard input	standard output
2 378 2020	2 55 323 3 2002 11 7

Problem J. Space Gophers

Input file:	*standard input*
Output file:	*standard output*
Time limit:	20 seconds
Memory limit:	512 mebibytes

Space Gophers live on a strange, cube-shaped asteroid, whose every side has length exactly $N = 10^6$. Their geogophers... sorry... their geographers have divided the asteroid into $N \times N \times N$ microcubes, and introduced a (standard Cartesian) coordinate system – every microcube has three coordinates (x, y, z).

The Gophers' main occupation (and also favorite pastime) is tunnel digging. A lot of tunnels have been made in the asteroid – each one starts at one of the asteroid's outer walls and is a straight line of removed microcubes, perpendicular to that wall, all the way through to the other side. But after many years of work, today is a grand festival, and all digging is forbidden for now. Any Gopher who wants to visit their friends must move through empty space only. This makes navigation hard for poor Gophers, so they really need your help. Given a list containing some pairs of starting and ending positions, determine if the end can be reached from the start by only moving through the empty microcubes (the Gophers only move through space they have dug, never going outside the initial cube. This is probably some kind of agopheraphobia).

Input

The first line of input contains the number of test cases z $(1 \le z \le 6)$. The descriptions of the test cases follow.

The first line of each test case contains a positive integer n $(n \le 300\,000)$ – the number of tunnels. The next n lines describe tunnels. Each one contains a triple of integers, of the form either $(x, y, -1)$, or $(x, -1, z)$, or $(-1, y, z)$, where $1 \le x, y, z \le 10^6$ are some integers. A triple $(x, y, -1)$ means that a tunnel has been dug by removing all (x, y, t) microcubes for every integer t. Similarly, a triple $(x, -1, z)$ means digging through all (x, t, z) cubes for all t, and $(-1, y, z)$ denotes a tunnel in place of all (t, y, z) cubes.

The next line contains the number of queries q $(1 \le q \le 500\,000)$, and is followed by q lines containing six integers each. Those six integers $(x_1, y_1, z_1, x_2, y_2, z_2)$, each of them between 1 and 10^6, denote a Gopher asking for help finding a route from the cube (x_1, y_1, z_1) to the cube (x_2, y_2, z_2). This route must only visit empty cubes, and it must always move between adjacent cubes. We consider cubes adjacent if they share a common wall. It is guaranteed that all starting and ending positions are empty.

Output

For each test case, output answers to all queries. For each query, output "YES" if there is a desired route, "NO" if there is none.

Example

standard input	standard output

1
6
-1 1 1
3 -1 2
1 5 -1
5 5 -1
-1 5 5
-1 5 9
6
1 1 1 6 1 1
8 1 1 3 2 2
1 1 1 5 5 5
1 5 5 5 5 9
1 5 10 10 1 1
1 5 9 5 5 9
	YES
	YES
	NO
	YES
	NO
	YES

Problem K. To argue, or not to argue

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

You are a director of a very successful theatre. Above all, you like William Shakespeare, even despite his inclination for bloody endings. It was said about some of his plays – like "Hamlet" and "King Lear" – that if they had just one more act, it would be necessary to start murdering people from the first rows of the audience.

Right now, you are close to developing a grudge for Shakespeare for not including this final act. It is because of the $2k$ people that have just come to your theatre. These are k pairs of celebrities – football players, models, YouTube streamers – who seem not to fully grasp the idea of theatre plays. Each pair is very likely to start a heated argument during the play, disrupting the performance entirely. But there is a solution – it is up to you to assign seats to people, and if a pair is not given adjacent seats, fight is much less likely.

The auditorium consists of n rows with m seats in each one. Some places are already booked by "normal" viewers, whom you do not want to reseat. There are k pairs of celebrities, and to every celebrity you must assign a seat, such that no pair occupies two adjacent spots (we consider two seats *adjacent* only if they share a common side, i.e. one is next to or behind the other). To cheer yourself up, compute the total number of ways you can do it – it is usually a very large number, so it is enough to compute its remainder modulo $10^9 + 7$. Two assignments are considered distinct if any celebrity is given a different seat. Please note that we distinguish all the celebrities (consider them **not** identical).

Input

The first line of input contains the number of test cases z $(1 \leq z \leq 100)$. The descriptions of the test cases follow.

The first line of each test case contains three positive integers n, m, k $(1 \leq n \cdot m \leq 144,\ 1 \leq k \leq mn/2)$ – the number of rows, seats in a row, and celebrity pairs. The next n lines describe the rows – each one is a string of characters '`X`' and '`.`', where '`.`' denotes a free seat, '`X`' – an occupied (unavailable) seats. You may assume that there are at least $2k$ free seats.

Output

For each test case, output a single number – the number of possible assignments of seats to celebrities such that no pair is given adjacent seats, modulo $10^9 + 7$.

Examples

standard input	standard output
2 2 2 2 .. .. 4 4 3 X.X. .... .X.. ...X	8 347040

Note

In the first example, all ways of assigning seats are presented below ('`A`' and '`a`' denote seats assigned to the first pair, '`B`' and '`b`' — to the second):

```
AB   Ab   ab   aB   BA   bA   ba   Ba
ba   Ba   BA   bA   ab   aB   AB   Ab
```

Problem L. Wizards Unite

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Greetings, Young Wizard! Your adventure begins here. But before you start, you have to prove that you are worthy. Shall you pass the test?

You have a bunch of old chests, each likely containing magical wonders of the wizarding world. You'd like to open them all by using keys that you have at your disposal.

In your key cabinet you have one gold key and k silver keys. Any key can open any chest, but each silver key can only be used once, while the gold key can be used multiple times. For each chest, you know how long it takes you to open it. You cannot use a single key to open several chests simultaneously. If you start opening a chest with some key, you have to wait till the chest is opened to reuse the key (of course, the key can be reused only if it's the gold one). On the other hand, you can use different keys simultaneously to open different chests, so it's possible that you are opening several chests at the same time (the described mechanics of opening chests with gold and silver keys actually exists in the game "Wizards Unite". It is allowed to use the insights gained by solving this problem when playing the game).

What is the minimum time needed to open all the chests?

Input

The first line of input contains the number of test cases z. The descriptions of the test cases follow.

The first line of each test case contains two integers n and k ($0 \le k < n \le 10^5$) – the number of chests and the number of silver keys.

The second line of each test case contains n integers t_i ($0 \le t_i \le 10^9$). Each t_i is the time needed to open the i-th chest.

The total number of chests in all test cases does not exceed 10^6.

Output

For each test case, output a single line which contains a single integer – the minimum time you need to open all the chests.

Example

standard input	standard output

2
3 1
1 3 2
3 2
5 5 5
	3
	5