Problem A: Assembly Required

Princess Lucy broke her old reading lamp, and needs a new one. The castle orders a shipment of parts from the Slick Lamp Parts Company, which produces interchangable lamp pieces.

There are m types of lamp pieces, and the shipment contained multiple pieces of each type. Making a lamp requires exactly one piece of each type. The princess likes each piece with some value, and she likes a lamp as much as the sum of how much she likes each of the pieces.

You are part of the castle staff, which has gotten fed up with the princess lately. The staff needs to propose k distinct lamp combinations to the princess (two lamp combinations are considered distinct if they differ in at least one piece). They decide to propose the k combinations she will like the least. How much will the princess like the k combinations that the staff proposes?

Input

The first line of input contains a single integer T ($1 \leq T \leq 10$), the number of test cases. The first line of each test case contains two integers m ($1 \leq m \leq 100$), the number of lamp piece types and k ($1 \leq k \leq 100$), the number of lamps combinations to propose. The next m lines each describe the lamp parts of a type; they begin with n_i ($2 \leq n_i \leq 100$), the number of pieces of this type, followed by n_i integers $v_{i,1}, \ldots, v_{i,n_i}$ ($1 \leq v_{i,j} \leq 10{,}000$) which represent how much the princess likes each piece. It is guaranteed that k is no greater than the product of all n_i's.

Output

For each test case, output a single line containing k integers that represent how much the princess will like the proposed lamp combinations, in nondecreasing order.

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

Explanation

In the first case, there are four lamp pieces, two of each type. The worst possible lamp has value $1 + 1 = 2$, while the second worst possible lamp has value $2 + 1 = 3$.

Problem B: Bulbs

Greg has an $m \times n$ grid of Sweet Lightbulbs of Pure Coolness he would like to turn on. Initially, some of the bulbs are on and some are off. Greg can toggle some bulbs by shooting his laser at them. When he shoots his laser at a bulb, it toggles that bulb between on and off. But, it also toggles every bulb directly below it, and every bulb directly to the left of it. What is the smallest number of times that Greg needs to shoot his laser to turn all the bulbs on?

Input

The first line of input contains a single integer T $(1 \leq T \leq 10)$, the number of test cases. Each test case starts with a line containing two space-separated integers m and n $(1 \leq m, n \leq 400)$. The next m lines each consist of a string of length n of 1s and 0s. A 1 indicates a bulb which is on, and a 0 represents a bulb which is off.

Output

For each test case, output a single line containing the minimum number of times Greg has to shoot his laser to turn on all the bulbs.

Sample Input	Sample Output
2 3 4 0000 1110 1110 2 2 10 00	1 2

Explanation

In the first test case, shooting a laser at the top right bulb turns on all the bulbs which are off, and does not toggle any bulbs which are on.

In the second test case, shooting the top left and top right bulbs will do the job.

Problem C: Card Collecting

Lately, a variety of free-to-play collectible card games have become popular. These card games usually have some collection of n cards that the player wants to collect. The player is first given a starter pack of s cards, which is guaranteed to not contain any duplicates (they are all unique). The player can then start acquiring new card(s) in two different ways:

1. Trade any set of d cards with a card of the player's choice. This can be done very quickly, and for this problem, we will assume that it takes no time.
2. Play games for an hour, and win a pack of k cards with probability p_i, where i is the number of distinct cards that the player currently holds. The pack consists of cards that are independently and uniformly at random from the entire collection of n cards (so it may or may not have duplicates).

The advantage of having a larger collection, of course, is that a player has a higher chance of winning a pack, and hence earns random cards more quickly.

Stingy Larry plays card games of this type all the time. Larry just got a copy of such a game (and is just about to open his starter pack of s cards), and wants to complete a full collection. Assuming that Larry manages his collection optimally, what is the shortest expected time to complete the collection that he can achieve?

Input

The first line of the input contains a single integer T ($1 \leq T \leq 10$), the number of test cases. The first line of each test case contains four integers n ($1 \leq n \leq 100$), the total number of cards in the collection, s ($0 \leq s \leq n$), the starting number of cards, k ($1 \leq k \leq 10$), the number of cards in a pack, and d ($1 \leq d \leq 100$), the number of cards that must be traded for a new card. The next line of the test case then contains $n+1$ space-separated real numbers $p_0, p_1, \ldots, p_n$ ($0.01 \leq p_i \leq 1$), which represent the probabilities of Larry winning a pack after an hour of play. You may assume the p_i's are nondecreasing.

Output

For each test case, output the shortest expected time for Larry to complete the collection in hours. An answer is considered correct if its absolute or relative error is at most 10^{-7}.

Sample Input	Sample Output
2 1 0 1 1 0.25 1.0 10 2 5 5 0.01 0.1 0.2 0.3 0.4 0.5 0.5 0.5 0.5 0.5 1.0	4.000000000 10.185962459

Explanation

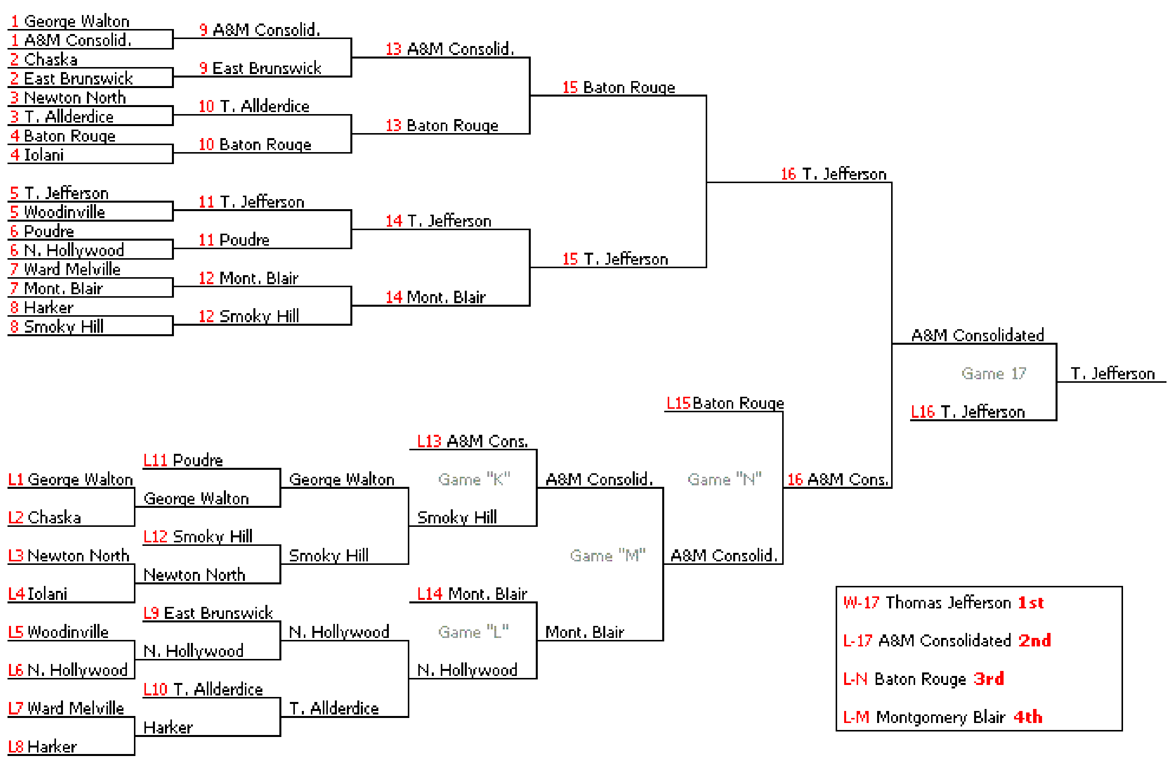
In the first test case, Larry will need to play an average of four hours to win the single card he needs to complete his collection.

Problem D: Double Elimination

Your friend J loves fighting games and plays them night and day. Everytime you try to play with J, he always wins in a crushing victory. You eventually suggest that J enter a fighting tournament, where he can play with people closer to his level. J decides to enter the Street League Pixel Championship, which is a bracketed double-elimination tournament.

Bracketed double-elimination tournaments work as follows. There are two brackets: the winners bracket and the losers bracket. Everyone begins in the winners bracket. Each round, players left in the winners bracket are paired and play matches. Winners remain in the winners bracket and losers drop down to the losers bracket.

Rounds of the losers bracket consist of a minor stage followed by a major stage. Suppose that at the beginning of a round, the losers bracket contains x players. In the minor stage, the x players in the losers bracket are paired and play matches. The $x/2$ winners from these matches remain in the losers bracket, and the other $x/2$ players are eliminated from the tournament. When $x/2$ players drop down from the winners bracket into the losers bracket, the major stage begins, where the new pool of x players are paired and play matches. The $x/2$ winners of this stage remain in the losers bracket, and the other $x/2$ players are eliminated from the tournament. This continues until there is only a single player in each bracket. These two players then play a match. If the player from the winners bracket wins, they win the tournament. Otherwise, the player from the player from the winners bracket drops down to the losers bracket and they play a final match (grand finals). The winner of that match wins the tournament.



Since you are a good friend, you watch all of J's games. J wins w games in the winner's bracket and ℓ games in the loser's bracket. What is J's final rank in the tournament? Participants are ranked by when they are eliminated from the tournament. Everyone eliminated at the same time is considered to be in a tie for the best possible rank they could all be. The winner of the tournament is rank one.

Input

The first line of input contains a single integer T ($1 \leq T \leq 10{,}000$), the number of test cases. The next T lines of input each contain three integers k ($0 \leq k \leq 30$), w, ℓ, representing a valid double-elimination tournament with 2^k competitors where J wins w games in the winner's bracket and ℓ games in the loser's bracket.

Output

For each double-elimination tournament, output a single line with a single integer giving J's rank.

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

Explanation

In the first tournament, there are four competitors. J wins his first game in the winners bracket, after which there are two players in the winners bracket and two players in the losers bracket. J loses his next game in the winners bracket, dropping down to the losers bracket. There is now one player in the winners bracket and two players in the losers bracket. J proceeds to beat the other player in the losers bracket to make it to the finals, where he defeats the winners bracket player twice to take the tournament.

In the second tournament, there are four competitors. J wins two games in the winners bracket to make it to the finals, where he wins to take the tournament.

In the third tournament, there are eight competitors. Believe or not, J drops out immediately, as does one other player, making the two tied for rank seven.

Problem E: Election of Evil

Dylan is a corrupt politician trying to steal an election. He has already used a mind-control technique to enslave some set U of government representatives. However, the representatives who will be choosing the winner of the election is a different set V. Dylan is hoping that he does not need to use his mind-control device again, so he is wondering which representatives from V can be convinced to vote for him by representatives from U.

Luckily, representatives can be persuasive people. You have a list of pairs (A, B) of represenatives, which indicate that A can convice B to vote for Dylan. These can work in chains; for instance, if Dylan has mind-controlled A, A can convince B, and B can convince C, then A can effectively convince C as well.

Input

The first line contains a single integer T ($1 \leq T \leq 10$), the number of test cases. The first line of each test case contains three space-separated integers, u, v, and m ($1 \leq u, v, m \leq 10{,}000$). The second line contains a space-separated list of the u names of representatives in U. The third line contains a space-separated list of the v names of representatives from V. Each of the next m lines contains a pair of the form `A B`, where `A` and `B` are names of two representatives such that `A` can convince `B` to vote for Dylan. Names are strings of length between 1 and 10 that only consists of lowercase letters (`a` to `z`).

Output

For each test case, output a space-separated list of the names of representatives from T who can be convinced to vote for Dylan via a chain from S, in alphabetical order.

Sample Input	Sample Output
2 1 1 1 alice bob alice bob 5 5 5 adam bob joe jill peter rob peter nicole eve saul harry ron eve adam joe chris jill jack jack saul	bob peter saul

Explanation

In the second test case, Jill can convince Saul via Jack, and Peter was already mind-controlled.

Problem F: Flight Plan

You work for SLPC Airlines, a small airline company that has just found a critical bug in its flight planner. The bug causes flights to take paths that keep either latitude or longitude constant. For example, to route a plane from $(0, 0)$ to $(10, 10)$, the flight planner would send the plane from $(0, 0)$ to $(0, 10)$ in a path keeping the latitude constant, and then from $(0, 10)$ to $(10, 10)$ in a path keeping the longitude constant.

You want to know how much the company has been losing from this bug. As a first step, you have the latitude/longitude coordinates of various cities and want to compute the shortest possible flight distance between them and the bugged flight distance between them.

Assume that the Earth is a perfect sphere with radius equal to 6,371 km, and that SLPC aircraft fly at a negligible height above the surface of the Earth.

Latitude and Longitude Reminder

The latitude of a point on the Earth's surface is the angle between the plane of the equator and the line going through the center of the Earth and that point. Latitudes range from 90°S (the South Pole) to 90°N (the North Pole).

The longitude of a point on the Earth's surface is the angle between a plane containing the Prime Meridian and a plane containing the North Pole, the point in question, and the South Pole (a meridian is half of a great circle from the North Pole to the South Pole; the Prime Meridian is a meridian chosen to have longitude 0). Latitudes range from 180°W to 180°E.

Input

The first line of input contains a single integer T ($1 \leq T \leq 10{,}000$), the number of test cases. Each test case is a single line with four real numbers a_1, b_1, a_2, b_2 ($|a_1|, |a_2| \leq 90$, and $|b_1|, |b_2| \leq 180$). These numbers form a pair of latitude/longitude coordinates (a_1, b_1) and (a_2, b_2), in degrees. Positive latitude coordinates represent points north of the equator; negative, south. Positive longitude coordinates represent points east of the prime meridian; negative, west.

Output

For each test case, output a single line giving the shortest possible flight distance between them followed by the bugged flight distance between them. An answer is considered correct if its absolute or relative error is at most 10^{-7}.

Sample Input	Sample Output
2 -90 0 90 0 0 0 10 10	20015.08679602 20015.08679602 1568.52055680 2223.89853289

Explanation

In the first case, the bugged flight planner happens to pick the best route between the two points; both distances are exactly half of a great circle.

In the second test case, the bugged flight planner does happen not pick the best route, so the answers differ.

Problem G: Ground Defense

You are a denizen of Linetopia, whose n major cities happen to be equally spaced along an east-west line. In fact, they are often numbered in order from 1 to n, where 1 is the westmost city and n is the eastmost city. Linetopia was a lovely place to live until forces from neighboring Trapez invaded. As part of Linetopia's Shielding Lives and Protecting Citizens initiative, you have been called upon to process information about Trapezoid troop movements so we can determine which cities have been hardest hit and know where to send reinforcements.

Linetopia intelligence has discovered that the Trapezoid forces are attacking in the following pattern. They are sending massive aircraft to drop troops on Linetopia cities. Each aircraft starts at some city i, dropping s soldiers. The aircraft then proceeds to fly either east or west. Each time it flies over another city, it drops a more soldiers than it dropped on the previous city it passed. After performing d drops, the aircraft returns to Trapez to resupply.

You will be receiving intel updates that inform you of the specs of each Trapezoid aircraft passing over Linetopia. You want to answer queries that ask how many Trapezoid troops have been dropped on a particular city. Are you up to the task?

Input

The first line of input contains a single integer T ($1 \leq T \leq 10$), the number of test cases. The first line of each test case contains two integers: m ($1 \leq m \leq 100{,}000$), the number of updates and queries and n ($1 \leq n \leq 500{,}000$), the number of cities in Linetopia.

The next m lines of input are either updates or queries. Update lines begin with a capital `U`, then contain either a capital `E` (east) or `W` (west) to indicate direction, and then contain four integers i ($1 \leq i \leq n$), s ($1 \leq s \leq 10{,}000$), a ($0 \leq a \leq 10{,}000$), and d ($1 \leq d \leq n$). These integers signify the starting city, the starting number of soldiers, the increase in soldiers per city, and the number of drops, respectively. You can assume d never results in an aircraft flying to the west of city 1 or to the east of city n.

Query lines begin with a capital `Q`, and then contain a single integer i ($1 \leq i \leq n$) indicating the city being queried.

Output

For each query in the input, output a single line containing the number of Trapezoid troops dropped in that city.

Sample Input	**Sample Output**

<pre>1
8 3
U E 1 5 2 3
Q 1
Q 2
Q 3
U W 3 10 10 2
Q 1
Q 2
Q 3</pre>
	<pre>5
	7
	9
	5
	27
	19</pre>

Explanation

Two aircrafts fly over Linetopia. The first starts at city 1 and heads east. It drops 5 soldiers on city 1, 7 soldiers on city 2, and 9 soldiers on city 3. The second starts at city 3 and flies west. It drops 10 soldiers on city 3 and 20 soldiers on city 2.

Problem H: Hunter's Apprentice

When you were five years old, you watched in horror as a spiked devil murdered your parents. You would have died too, except you were saved by Rose, a passing demon hunter. She ended up adopting you and training you as her apprentice.

Rose's current quarry is a clock devil which has been wreaking havoc on the otherwise quiet and unassuming town of Innsmouth. It comes out each night to damage goods, deface signs, and kill anyone foolish enough to wander around too late. The clock devil has offed the last demon hunter after it; due to its time-warping powers, it is incredibly agile and fares well in straight-up fights.

The two of you have spent weeks searching through dusty tomes for a way to defeat this evil. Eventually, you stumbled upon a relevant passage. It detailed how a priest managed to ensnare a clock devil by constructing a trap from silver, lavender, pewter, and clockwork. The finished trap contained several pieces, which must be placed one-by-one in the shape of a particular polygon, in counter-clockwise order. The book stated that the counter-clockwise order was important to counter the clock devil's ability to speed its own time up, and that a clockwise order would only serve to enhance its speed.

It was your responsibility to build and deploy the trap, while Rose prepared for the ensuing fight. You carefully reconstruct each piece as well as you can from the book. Unfortunately, things did not go as planned that night. Before you can finish preparing the trap, the clock devil finds the two of you. Rose tries to fight the devil, but is losing quickly. However, she is buying you the time to finish the trap. You quickly walk around them in the shape of the polygon, placing each piece in the correct position. You hurriedly activate the trap as Rose is knocked out. Just then, you remember the book's warning. What should you do next?

Input

The first line of input contains a single integer T ($1 \leq T \leq 100$), the number of test cases. The first line of each test case contains a single integer n ($3 \leq n \leq 20$), the number of pieces in the trap. Each of the next n lines contains two integers x_i and y_i ($|x_i|, |y_i| \leq 100$), denoting the x and y coordinates of where the ith piece was placed. It is guaranteed that the polygon is simple; edges only intersect at vertices, exactly two edges meet at each vertex, and all vertices are distinct.

Output

For each test case, output a single line containing either `fight` if the trap was made correctly or `run` if the trap was made incorrectly.

Sample Input	Sample Output
2 3 0 0 1 0 0 1 3 0 0 0 1 1 0	fight run

Explanation

In the first case, you went around the polygon in the correct direction, so it is safe to fight the clock devil and try to save Rose.

In the second case, you messed up, and it is time to start running. Sorry Rose!

Problem I: Ingenious Lottery Tickets

Your friend Superstitious Stanley is always getting himself into trouble. This time, in his Super Lotto Pick and Choose plan, he wants to get rich quick by choosing the right numbers to win the lottery. In this lottery, entries consist of six distinct integers from 1 to 49, which are written in increasing order. Stanley has compiled a list of winning entries from the last n days, and is going to use it to pick his winning numbers.

In particular, Stanley will choose the six numbers that appeared the most often. When Stanley is breaking ties, he prefers smaller numbers, except that he prefers seven to every other number. What is Stanley's entry?

Input

The first line of input contains a single integer T ($1 \leq T \leq 100$), the number of test cases. The first line of each test case contains a single integer n ($1 \leq n \leq 1{,}000$), the number of winning entries that Stanley compiled. The next n lines each contain a lottery entry as described above.

Output

For each test case, output a single line containing Stanley's entry.

Sample Input	Sample Output
2 3 1 2 3 4 5 6 4 5 6 7 8 9 7 8 9 10 11 12 3 1 2 3 4 5 6 4 5 6 7 8 9 1 2 3 7 8 9	4 5 6 7 8 9 1 2 3 4 5 7

Explanation

In the first test case, the numbers 4 through 9 appear twice each, while all other numbers appear at most one time.

In the second test case, all numbers 1 through 9 appear twice each. The tiebreaking rule means Stanley prioritizes picking 7 and then the five smallest numbers.

Problem J: Jurisdiction Disenchantment

The Super League of Paragons and Champions (SLPC) has been monitoring a plot by a corrupt politician to steal an election. In the past week, the politican has used a mind-control technique to enslave the n representatives responsible for choosing the election's winner. Luckily, the SLPC has managed to recruit you and hence has access to your power to break mind-control. You are able to break mind-control in an axis-aligned rectangle. Unfortunately, your power comes at a steep cost; you get a headache the next day proportional to the size of the rectangle. You do not even want to risk or think about what would happen if you tried to use your power multiple times in one day.

You have done your research and you know the position that each representative will be standing when the votes are about to be cast. You need to free enough representatives to prevent the politician from having a majority (strictly more than one-half) vote. What is the area of the smallest axis-aligned rectangle that you can affect to do this?

Input

The first line of input contains a single integer T ($1 \leq T \leq 10$), the number of test cases. The first line of each test case contains a single integer n ($1 \leq n \leq 299$, n is odd), the number of representatives. Each of the next n lines of input contains two integers, denoting the x and y coordinates of a representative. It is guaranteed that all coordinates are between −10,000 and +10,000.

Output

For each test case, output a single line containing the area of the smallest axis-aligned rectangle containing more than $n/2$ of the representatives.

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

Explanation

In the first case, a rectangle containing a single point has an area of 0.

In the second test case, the rectangle needs to include at least two points. There are two smallest possible rectangles; one includes $(0, 0)$ and $(1, 4)$ and the other includes $(1, 4)$ and $(3, 2)$. In either case, the area is 4.

Problem K. Mixing Bowls

You are following a recipe to create your lunch.

The recipe is a mixture made by combining ingredients together in a bowl. Each ingredient will be either:

Another mixture which you must make first in a separate bowl; or A basic ingredient you already have in your kitchen, which can be added directly. To make a mixture, you need to have all its ingredients ready, take an empty bowl and mix the ingredients in it. It is not possible to make mixtures by adding ingredients to an already-existing mixture in a bowl.

For example, if you want to make `CAKE` (a mixture) out of `CAKEMIX` (a mixture) and `lies` (a basic ingredient), then you must first make `CAKEMIX` in its own bowl, then add the `CAKEMIX` and `lies` to a second bowl to make the `CAKE`.

Once you have used a mixture as an ingredient and emptied the bowl it was prepared in, you can re-use that bowl for another mixture. So the number of bowls you need to prepare the recipe will depend on the order in which you decide to make mixtures.

Determine the minimum number of bowls you will need.

Input

The first line will contain an integer N ($1 \le N \le 1000$), the number of mixtures.

Following N lines, one for each mixture, contain:

- One string giving the mixture name;
- An integer M ($2 \le M \le 10$), the number of ingredients in this mixture;
- M strings, giving the names of each of the ingredients of this mixture.

The tokens on one line will be separated by single spaces.

The first mixture is the recipe you are making.

The names of mixtures are strings of between 1 and 20 UPPERCASE letters.

The names of basic ingredients are strings of between 1 and 20 lowercase letters.

Each mixture is used in exactly one other mixture, except for the recipe, which is not used in any other mixture. Each ingredient will appear at most once in the ingredient list for a mixture. No mixture will (directly or indirectly) require itself as an ingredient.

Output

Output one number Y — the minimum number of mixing bowls required.

Examples

standard input
3 SOUP 3 STOCK salt water STOCK 2 chicken VEGETABLES VEGETABLES 2 celery onions

standard output
2

standard input
5 MILKSHAKE 4 milk icecream FLAVOR FRUIT CHOCOLATE 2 cocoa syrup FLAVOR 2 SPICES CHOCOLATE FRUIT 2 banana berries SPICES 2 nutmeg cinnamon

standard output
3

Notes

In the first example, to satisfy your craving for `SOUP`, you follow these steps:

1. Make `VEGETABLES` by mixing celery and onions in a bowl.
2. Make `STOCK` in a second bowl by mixing `chicken` and `VEGETABLES` from the first bowl. The first bowl becomes empty.
3. Make `SOUP` in the first bowl by mixing `STOCK`, `salt` and `water`.

In the second example, you have a choice of whether to make `FLAVOR` or `FRUIT` first before mixing them with `milk` and `icecream` to make `MILKSHAKE`.

If we make `FRUIT` first, we use four bowls:

1. Make `FRUIT` in a bowl by mixing `banana` and `berries`.
2. Make `SPICES` in a second bowl by mixing `nutmeg` and `cinnamon`, and `CHOCOLATE` in a third bowl by mixing `cocoa` and `syrup`. (In either order)
3. Make `FLAVOR` in a fourth bowl by mixing `SPICES` and `CHOCOLATE`.
4. Make `MILKSHAKE` in the second or third bowl by mixing `FRUIT`, `FLAVOR`, `milk` and `icecream`.

However if we make `FRUIT` after `FLAVOR`, we use three bowls:

1. Make `SPICES` in a bowl by mixing `nutmeg` and `cinnamon`, and `CHOCOLATE` in a second bowl by mixing `cocoa` and `syrup`. (In either order)
2. Make `FLAVOR` in a third bowl by mixing `SPICES` and `CHOCOLATE`.
3. Make `FRUIT` in the first bowl by mixing `banana` and `berries`.
4. Make `MILKSHAKE` in the second bowl by mixing `FRUIT`, `FLAVOR`, `milk` and `icecream`.

Problem L. Sympathetic Tables

Let's call table $N \times M$ consisting of 0-s and 1-s *sympathetic* if and only if each square 2×2 of this table contains at least one 0 and at least one 1.

You are given two *sympathetic* tables A and B. You task is to answer to the question — is it possible to get the table B from the table A if in one step you are able to change the value of one element of the table (change 0 to 1 or 1 to 0). All intermediate tables must be *sympathetic*. The number of the steps in your solution must not exceed $7 \cdot M \cdot N$.

Input

The first line contains two integers N and M ($1 \le N, M \le 1000$).

Following N lines contain the descriptions of the table A (M numbers in each line separate by a spaces). Following N lines contain the descriptions of the table B in the same format.

Output

If it is not possible to get the table B from the table A with given restrictions print the number `-1`.

In the other case the first line of output must contains single integer K — the number of steps. Following K lines must contain the descriptions of steps — one step per line. Each step described by coordinates (i, j) of element which must be changed, where i is the number of row $(1 \le i \le N)$ and j is the number of column $(1 \le j \le M)$. Numbers in lines must be separated by a space.

Examples

standard input	standard output
2 2 0 0 1 0 0 1 0 0	2 1 2 2 1

Problem M. Cows

In the village Miniyaroslavka cows are grazing on the lawns, which connected by a walkways. On each lawn at least one cow is grazing. For any pairs of lawns there is exactly one way to go from the one lawn the other. Each walkway is bi-directional.

The main farmer of the village wants to build two cowsheds for his cows. Each cow will return to the cowshed which is nearer to her lawn (if the distances are equal cow will return to any cowshed). Because of that you need to choose such places to build cowsheds that the maximum distance, which cow will overcome to return to her cowshed, be as least as possible.

Input

The first line contains integer N $(2 \le N \le 100000)$ — the number of lawns.

Following $N - 1$ lines contain the description of the walkways. Each walkway is described by a three positive integers (a, b, w), where a and b — the lawns, which this walkway connects (the lawns are numbered from 1) and w is a length of the walkway in miles $(w \le 10000)$.

Output

Print two numbers u and v — the number of lawns where the main farmer must build the cowsheds.

Examples

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