Problem A. Freestyle

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

Freestyle snowboarding is one of the most entertaining winter sports, in which athletes compete while performing tricks of various difficulty.

Freestyle is a creative and fairly young sport that does not have established training methods. Many professional freestylers do not have a coach at all — they learn new tricks while gathering at snow parks with their friends and experimenting to their heart's content.

Two snowboarders developed their own method for shared practices. Before beginning practice, they use some method to create a sequence of N tricks, each of which must be performed once during any round. Each trick is assigned a unique identifier A_i, which corresponds to the difficulty level, from easiest — 1, to hardest — N.

The snowboarders take their rounds in turn and each round must be harder than the previous. To assure this, the snowboarder who finished a round must change the order of tricks performed by his opponent in the previous round such that the sequence becomes lexicographically higher — in other words, for a position k, he must meet the conditions $Q_k > P_k$ and $Q_i = P_i$ when $i < k$, where Q and P are the current and previous sequences of tricks, respectively. For first round as «previous» selected sequence is used.

Changing the order of performing tricks relative to the previous round is limited by the following condition: the snowboarder must select one sequence of tricks in a row and put them in the opposite order. In addition, the selected sequence must have the length $4 \times x + 2$ or $4 \times x + 3$, where x is any non-negative integer.

If the snowboarder is not able to change the order of tricks before the round such that the round becomes more difficult than the previous one, the practice session ends. Each snowboarder wants to perform the very last round and prove that he is better than his opponent, so both participants act optimally when selecting trick sequences for each round. You must determine which of the participants will perform the last round.

Input

The first line sets the number of tricks in N rounds ($1 \le N \le 100$). The second line sets N comma-separated trick IDs A_i ($1 \le A_i \le N$, $A_i \ne A_j$ when $i \ne j$) in the same sequence in which they were selected before the first round.

Output

If the last round is performed by the snowboarder who performed the first round, output $First$; otherwise, output $Second$.

Examples

Standard input	Standard output
3 1 2 3	First
4 4 2 1 3	Second

Problem B. Checkout lines

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

Many fans of downhill skiing and snowboarding are not able to take a vacation in the middle of winter to go skiing in the mountains. Luckily, Russia has holidays during the cold time of year — the New Year holidays, February 23rd, and March 8th. During any of these holidays, all the local ski resorts are filled with crowds of people who have come to ski for just a few days, and huge lines form at the ticket counters, rental shops, and lifts from early morning.

N people gathered at one of the rental shops prior to opening, forming two different lines of the length N: line A to get ski boots, and line B to get skis. However, each person had one place in each of the lines.

The lines usually move symetrically, because getting boots and skis requires the same amount of time. But if at some point it is one person's turn to get both boots and skis, he can't get both at once, and all the other people must wait.

To avoid this type of delay, the resort administration has the goal to change the order of people in line B such that the position of any particular person in line A differs from his position in line B.

After changes are made, some people might move back in line B, which, of course, causes them displeasure. Assume the position of person i in line B before the changes was p_i, and after the changes it was q_i. Then the deviation of the position will be the difference $q_i - p_i$. A person's position is considered to have worsened if the deviation is positive.

To minimize the number of disgruntled customers, the administrator wants to reach this goal in a way that the maximum deviations for the worse (i.e. maximum of $q_i - p_i$ for all i for which $q_i >= p_i$) are reduced to a minimum.

Input

The first line sets the number of people N ($1 \le N \le 10^5$).

The second line sets N numbers A_i ($1 \le A_i \le N$, $A_i \ne A_j$ when $i \ne j$) — the order of people in line A.

The third line sets N numbers B_i ($1 \le B_i \le N$, $B_i \ne B_j$ when $i \ne j$) — the order of people in line B.

Output

If there is a solution, output the minimum possible maximum deviation in position for the worse in the first line, and output the resulting order of people in line B in the second line. If there are multiple solutions, output any of them.

If there is no solution, print -1.

Examples

Standard input	Standard output
2 1 2 2 1	0 2 1
3 1 2 3 3 2 1	1 2 3 1

Problem C. Heli-ski (Division 1 Only!)

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

Heli-skiing is an extreme type of downhill skiing, in which mountain slopes are accessed by helicopter, and the descent is made on untouched virgin snow, far from civilization.

A team of free-riders that were flying in a helicopter over a previously unexplored mountain spine noticed, among the impassable cliffs, a large expanse of virgin snow ideal for a descent. The weather made it impossible for them to land there that day, so they decided to take some pictures of the area so they could thorougly plan their route and conquer the slope on the following day.

Upon returning to camp, the free-riders printed N photographs, each with the same height of H and a width of W_i, lined them up in a row on the table, and went to rest. When they gathered in the evening to discuss their upcoming ski route, they discovered that the order of the photos was mixed up.

To get an approximate idea of the size of the area, they decided to change the order of the photos in the row to get the largest possible area of the maximum rectangular area after taping neighboring photos together. However, while rearranging the photos, they cannot be turned over, photos that are located next to each other are taped together along the entire height, and the edges of the discovered area must be parallel to the sides of the photos.

Input

The first line sets comma-separated numbers N and H ($1 \le N, H \le 10^5$).

This is followed by N photo descriptions. In the first line of each description, the width of the photo is set W_i ($1 \le H \times \sum_{i=1}^{n} W_i \le 10^5$). The next H lines of W_i characters set a photograph, where 0 characters correspond to sections that are good for skiing down, while 1 characters represent sections that have cliffs. It is guaranteed, that atleast one section is good for skiing.

Output

In the first line, you must output the rearrangement of N numbers from 1 to N that allow us to get the largest possible rectangular area that is suitable for skiing.

In the second line, you must output the coordinates of the upper-left and lower-right corners of the found section after taping the photos in the order set by the rearrangement. After taping them together, the upper-left corner of the first photo has the coordinates $(1, 1)$, and the lower-right corner of the last photo has the coordinates $(H, \sum_{i=1}^{n} W_i)$. If there are multiple optimal solutions, output any of them.

Examples

Standard input	Standard output
3 4 4 1001 0000 0010 1001 3 000 010 000 011 2 00 10 01 00	2 1 3 2 3 3 5

Note

In the first test example, after taping them together, the photos are positioned as follows (the unknown section is highlighted in bold):

000100100

01**000**0010

00**000**1001

011100100

Problem D. Apres-ski

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

Apres-ski is translated directly from French as "after skis". This term actually covers the entire spectrum of entertainment available at mountain resorts that directly follows skiing — parties, clubs, excursions, shopping, and of course, gambling.

Two skiers, traveling in Europe for the first time, underestimated the prices at the local resorts, so after just a few days, their apres-ski was reduced to a meagre supper and the four walls of their hotel room. To somehow entertain themselves after skiing, they made up a new game.

Before the game begins, they take a piece of paper and write down a series of N different positive integers A_i. Players take turns. On a turn, a player can substract 1 from one of the numbers in the series. If the player's turn results in two identical numbers appearing in the series, one of them is crossed out. If the player's turn results in 0 appearing in the series, it is also crossed out. The player who is not able to take a turn is the loser, and his opponent is the winner.

Your task, knowning the series of numbers on the paper, is to determine the winner.

Input

The first line sets the length of the series N ($1 \le N \le 3$).

The second line sets N space-separated items in the series A_i ($1 \le A_i \le 10^{18}$). All the items in the series are different.

Output

If the player who goes first is the winner, output 1, otherwise output 2.

Examples

Standard input	Standard output
1 42	2
2 4 2	2
3 2 3 4	1

Problem E. Land in Krasnaya Polyana

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

The approaching 22nd Winter Olympic Games in Sochi have transformed the nearby village of Krasnaya Polyana into something unrecognizable, where skiing, sledding, and other competitions will be held. Over a short period, three new downhill ski complexes have been constructed, the transportation infrastructure has been completely redesigned, hundreds of new hotels have been built... And the construction is not even nearing an end.

For the construction of new hotels, the village administration has set off a large rectangular section of land, divided equally into $N \times M$ equal sectors, located in N rows and M columns. All of the sectors were put up for sale to private companies and each has a set price of C_{ij}.

There are two competing hotel chains — A and B. As often happens in big business, despite their being in competition with each other, both companies belong to the same owner. The owner wants to build N hotels of the A chain so that each row in the section has one A hotel, and M hotels of the B chain so that each column in the section has one B hotel. In addition, exactly one sector is needed for the construction of each hotel, and no two hotels can be built in the same sector.

Your task is to determine the minimal cost S that the owner will have to pay to purchase land for building the hotels according to the layout described.

Input

The first line sets the number of rows and columns in the section of land: N and M ($N \times M \leq 10^6$, $2 \leq N, M$).

The following N lines set the cost of sectors in rows, by M numbers C_{ij} ($1 \leq C_{ij} \leq 10^9$) in each.

Output

In the first line, output the minimal cost of purchasing the land S.

Examples

Standard input	Standard output
3 4 1 3 10 8 2 1 9 2 6 7 4 6	19

Note

In the test example, it is assumed that the A chain hotels are located in the sectors $(1, 1)$, $(2, 2)$ and $(3, 1)$, and the B chain hotels in the sectors $(2, 1)$, $(1, 2)$, $(3, 3)$ and $(2, 4)$.

Problem F. Transfer

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

The closer a hotel is located to the ski lifts, the more expensive it is to stay there — every skier knows this rule. It's obvious that walking several kilometers to the lifts is not most people's idea of fun, nor is taking public transit every morning with bulky equipment. But luckily for skiers, many inexpensive hotels located at a distance from the resorts have recently started offering their own transfer services.

A bus that is taking N passengers from the hotel to the lifts in the morning makes one stop near the ski equipment rental on its way. There is a probability of Pa_i that each passenger might be sleeping at the time of this stop — sleeping passengers do not wake up until the bus starts moving again. There is a probability of Pb_i that a passenger stays in the bus and stays awake during the time while the bus is stopped. There is a probability of Pc_i that a passenger leaves the bus at the rental shop.

After a while, the driver gets tired of waiting for the passengers who got out, and uses the microphone to ask "has everyone taken their seats?". Sleeping passengers cannot answer the question. Passengers who got off the bus do not hear the question and do not know that the bus is getting ready to leave. Each passenger who is awake in the bus answers "no" if one of his friends is not on the bus, but he does not answer if all his friends are there. If the driver hears at least one "no", he asks the passengers to call their friends on the phone and ask them to come to the bus. The passengers who answered "no" call their friends, who immediately return to the bus. After this, the driver asks the same question again using the microphone.

This process is repeated until none of the passengers answers "no" when the driver asks the question; in this case, the bus leaves, and anyone who did not return to the bus is left at the rental shop.

You must determine the expected value of the number of passengers left during the stop, if you know the list of friends of each of the passengers. It is known that friendships form a single tree structure, so each passenger is friends with at least one other passenger, and there are no cyclical friend relationships (if A and B are friends, and B and C are friends, then A cannot be friends with C). Note that none of the passengers who leaves the bus will return to the bus until one of their friends calls them from the bus.

Input

The first line sets the number of passengers N ($1 \le N \le 50000$). The following N lines set event probabilities for each passenger, as percentages: Pa_i, Pb_i, Pc_i ($0 \le Pa_i, Pb_i, Pc_i \le 100$, $Pa_i + Pb_i + Pc_i = 100$).

The following $N - 1$ lines describe the friendship trees: each line contains a single pair of friends u_i and v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$).

Output

The expected value of the number of passengers left behind at the stop. The answer must not differ from the standard output by more than 10^{-6}.

Examples

Standard input	Standard output
1 32 46 22	0.2200000000
2 32 46 22 54 23 23 1 2	0.2936000000
5 78 12 10 17 13 70 39 19 42 22 16 62 93 1 6 1 5 2 5 2 3 4 5	1.6404454080

Problem G. Building ski lifts

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

Over the last few years, several different ambitious ski resort construction projects have been started simultaneously in Russia. One of the most important stages of creating a new resort is researching the surrounding area and designing a project for ski tracks that will be the starting point for planning transportation and hotel infrastructure. Besides the ski track project, there must be provisions for the construction of ski lifts from the very beginning of the project.

A ski track project is made up of a set of M directional sections of track that connect certain pairs of N reference points on the slopes. For each reference point, we know its altitude H_i above sea level, and the direction of track sections goes from higher points to lower ones. A set of reference points is connected, meaning there is a path across track sections from each reference point to any other point, without taking direction into account.

The investors who are involved in building a new resort are interested in getting returns on their investment as quickly as possible. You have been asked to take a prepared ski trail project and develop a starting project for constructing simple rope tow lifts according to several criteria.

1. A lift can connect any two different reference points S_i and F_i, but only allows skiers to move in the direction from point S_i to point F_i, and must meet the conditions $H_{S_i} \le H_{F_i}$.

2. After lifts are built, it must be possible to get from each reference point to any other one by skiing in the direction of the track sections and/or using the lifts.

3. The total difference in altitude $Z = \sum_{i=1}^{K} H_{F_i} - H_{S_i}$ of the lifts being built must be the lowest possible.

4. From all the solutions that meet the first three conditions, you must choose the solution with the minimal number of K lifts in the project.

Input

The first line sets the number of reference points N ($1 \le N \le 100$) and the number of track sections M ($1 \le M \le 10000$).

The second line sets N numbers H_i ($0 \le H_i \le 10000$) - the altitude of the reference points. All H_i are pairwise distinct.

The following M lines set the track sections as pairs of reference point numbers that they connect: U_i and V_i ($1 \le U_i, V_i \le N$, $H_{U_i} > H_{V_i}$). The input data does not have any track sections that connect identical pairs of reference points.

Output

In the first line, output the total difference in altitude for the lifts Z.

In the second line, output the number of lifts in your project K.

In the following K lines, output the numbers of the reference points S_i and F_i that are connected by lifts.

If there are multiple possible solutions that meet the conditions, output any one of them.

Examples

Standard input	Standard output
4 3 10 8 6 0 1 2 2 3 2 4	10 2 4 3 3 1

Problem H. The lost key (Division 1 Only!)

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

A group of downhill skiers rented a cottage at a European ski resort. After returning from a day of skiing, they realized that they had lost the only key to the cottage that had been given to them.

They usually leave the key near the door, so that any member of the group returning from the slopes can easily get inside. But today, one of them accidentally took the key with him when he went skiing, and lost it somewhere on the slopes. When he realized that the rest of the group would have to spend hours freezing outside the cottage because of his mistake, the member who lost the key decided to stay out of sight until the next day and went to a night club, hoping that they would be able to contact the landlord and get a spare key.

However, none of the group's efforts to get a hold of the landlord resulted in success. In despair, they set off to search for the key on the slopes. To speed up the search, they decided to write down the tracks where the group members each said they had seen the guy who lost the key today. However, before they went out searching, they had to check the trail map to verify whether the information they had gathered was reliable or not.

The resort where they are staying is part of a large ski region where there are many ski resorts connected by a shared infrastructure. The trail map for the ski area is a connected set of N reference points connected by $N-1$ tracks, each marked with a marker c_i. Since the ski area includes multiple resorts, the markers on the tracks may be duplicated. In addition, the map that fell into the hands of the group does not indicate the altitude of the reference points, so when solving this task, you must assume that these are two-way tracks that can be skied in both directions.

Let's look at the shortest route on the ski tracks between the reference points u and v: P_1, P_2, P_3, $\cdots$, P_k, where $P_1 = u$, $P_k = v$ and k — the number of reference points in the shortest route. We'll write out the track markers in the order they are passed on the route: S_1, S_2, $\cdots$, S_{k-1}, where the S_i marker is on the tree edge (P_i, P_{i+1}). Then the sequence of S markers is a valid route, and the pair of reference points (u, v) is the position of the route on the trail map.

You are given a set of M sequences of P markers. For each sequence, you must verify whether it is a valid route, and if so, find one of its positions on the trail map.

Input

The first line contains a single integer N ($1 \le N \le 50000$) - the number of reference points on the trail map.

The following $N-1$ lines contain descriptions of tracks in the form of three numbers u_i, v_i and c_i ($1 \le u_i, v_i \le N$, $0 \le c_i \le 10^5$), where u_i, v_i are reference points connected by a track, and c_i is a marker on the track.

The next line has a single integer M ($1 \le M \le 50000$) – the number of sequences that must be checked.

Each of the following M lines contains a sequence P_i in the following format: length L_i ($1 \le L_i$) of the sequence and L_i items in the sequence $P_{i,j}$ ($0 \le P_{i,j} \le 10^5$). All numbers are integers and separated by a single space.

It is guaranteed that $M \le \sum_{i=1}^{M} L_i \le 50000$.

Output

Output M lines, one for each sequence P_i. If a sequence P_i is a valid route, output a pair of vertices u_i, v_i, describing the position of the route on the trail map. If there are several route positions, output any one of them. If the string P_i is not a valid route, output two zeros separated by a space.

Examples

Standard input	Standard output
6 1 2 1 2 3 7 1 6 2 6 4 3 6 5 3 14 1 1 2 1 2 2 2 1 4 7 1 2 3 5 7 1 2 3 3 2 3 3 2 1 1 3 3 2 1 4 3 2 1 7 3 2 1 7 3 7 1 7 3 2 3 2 1 7 1 5	2 1 2 6 6 2 3 4 0 0 4 5 0 0 4 2 4 3 6 3 0 0 0 0 3 2 0 0
4 1 2 2 1 4 1 2 3 3 9 2 2 3 2 3 2 1 2 1 3 1 4 2 1 1 2 1 2 3 1 2 3 3 3 2 1	1 3 3 1 2 1 3 2 0 0 0 0 4 2 4 3 3 4

Problem I. Up and down (Division 2 Only!)

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

When skiing at certain downhill ski resorts, you might find yourself thinking that you have to go up on the lifts more often than you get to go down on skis. The reason for this is usually poor infrastructure at the resort.

To get an objective estimate of the quality of resort infrastructure, some skiers track statistics of their navigation. The results are written in the string S: when skiing down a hill, the '$>$' symbol is added to the string, and when going up on ski lifts, the '$<$' symbol is added.

For a known string S, you must reconstruct the sequence of peaks H that a skier could have been on. The H sequence must have a length equal to $|S| + 1$ and consist of integers $0 \le H_i \le 10^9$. In addition, if position i in the string S has the '$>$' symbol, the condition $H_i > H_{i+1}$ must be met, and if it has the '$<$' symbol, the condition $H_i < H_{i+1}$ must be met. Of all the acceptable sequences, you must select the one that has the minimal number of different peaks H_i.

Input

A single string S ($1 \le |S| \le 10^5$), consisting only of the symbols '$<$' and '$>$'.

Output

The unknown sequence of peaks H.

If there are multiple solutions, output any of them.

Examples

Standard input	Standard output
$>>><<<$	`9 6 3 0 3 6 9`
$>>><><<$	`3 2 1 0 1 0 1 2`

Problem J. Ski equipment (Division 2 Only!)

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

Good equipment is essential to comfortable skiing. In addition to skis, poles, and boots, you will need a special ski jacket, pants, a mask, a helmet, gloves, and a number of other items as well.

Before your ski trip, you planned to purchase N diferent items in a sporting goods store, but you do not know the exact price of each of them. However, according to your friends, the price of the i-th item is an integer and is located on the segment $[L_i, R_i]$.

You must calculate the number of different combinations of P prices for items in which the price of each item satisfies the requirement $L_i \le P_i \le R_i$, and the bitwise sum $S = P_1 \oplus P_2 \oplus ... \oplus P_N$ (XOR operation) of all the purchased items is exactly equal to X.

Input

The first line contains two numbers, N and X ($2 \le N \le 8$, $0 \le X \le 10^{18}$).

The next N lines set pairs of numbers, L_i and R_i ($0 \le L_i \le R_i \le 10^{18}$).

Output

Output the unknown number of price combinations for the absolute value 1 000 000 007.

Examples

Standard input	Standard output
3 5 0 3 4 7 2 5	8