Problem A. BankCraft

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

Osya and his gang want to expropriate some money from a backstreet millionaire Koreyko. But there is a problem: Koreyko stores all his money in a bank. The bank uses public-key cryptography to authenticate its clients. Each client has his own public key which is a polynomial $P(x)$ over the field of integers modulo p, and a secret key which is a polynomial $Q(x)$ over the same field. The secret key is considered to be valid if there exists a polynomial $R(x)$ such that $P(x) \cdot Q(x) = 1 + x^m \cdot R(x)$ for a known integer m.

Osya knows the polynomial $P(x)$ and integers p (it is always equal to $7\,340\,033$) and m, but he doesn't know the secret key. He offers you a key to the house full of money for your help with finding the secret key. You can't reject such a generous proposition.

Input

The first line of input contains two integers: m and n ($1 \le m, n \le 10^5$). Here, n is the degree of $P(x)$. The second line contains $n+1$ integers a_i ($0 \le a_i \le p-1$) separated by spaces, which are the coefficients of $P(x)$. The i-th of them ($0 \le i \le n$) is the coefficient of x^i.

Output

If it is impossible to construct the required polynomial of degree less than m, output the message "`The ears of a dead donkey`" (without quotes). If a solution exists output m integers b_i ($0 \le b_i \le p-1$) which are the coefficients of $Q(x)$, separated by spaces. If there are multiple valid answers, output your favorite one.

Examples

standard input	standard input
2 1 1 2	1 7340031
4 2 1 0 1	1 0 7340032 0

Problem B. Cave Exploring

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

A cave in this problem looks like two polylines on the plane, one below the other, each containing N vertices. The first polyline is a floor of the cave while the second one is its ceiling. The coordinates of the i-th floor vertex $Floor_i$ are (i, f_i) and the coordinates of the i-th ceiling vertex $Ceiling_i$ are (i, c_i), where $0 \le i \le N-1$. The floor is strictly lower than the ceiling, and their polylines have no common points.

A light beam is an infinite straight strip with fixed width. The floor and the ceiling of the cave are considered to be transparent, which means that the beam goes through them without reflections or refractions.

You have to run the light beam in the cave in such a way that it is possible to fly through the cave (starting from some point on the segment $[Floor_0, Ceiling_0]$ and finishing at some point on the segment $[Floor_{N-1}, Ceiling_{N-1}]$) by some continuous trajectory, which is located completely inside the lightened area. The trajectory is allowed to touch the beam bounds, the floor and the ceiling of the cave, but it is not allowed to intersect any of them.

You can run the beam on the plane in any way you want, however you have to minimize its width. We are not asking you to find the way of running the beam, you just need to find its minimal possible width.

Input

The first line of input contains an integer N ($2 \le N \le 10^6$), the number of floor and ceiling vertices. The next line contains exactly N integers f_i which are the y-coordinates of floor vertices. The next line contains N integers c_i which are the y-coordinates of ceiling vertices ($0 \le f_i < c_i \le 10^8$).

Output

Output a single real number which is the minimum width of the beam that allows to fly through the cave. The answer must have an absolute or a relative error less than 10^{-9}.

Example

standard input	standard input
5 0 1 3 0 0 2 4 4 4 1	1.4552137502179978

Problem C. Girl vs Gopniks

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

A little girl named Telle is the main character of one of Lukyanenko's books. She is thirteen but she is used to face Gopniks (Russian common word for hooligans) very often. Sometimes Gopniks win the battles, sometimes the girl does. It is known that N Gopniks live in the city. There were K battles between Telle and Gopniks. For each battle, the set of Gopniks that attacked the girl and the winner of the battle are known.

Critics say that the events in the book are inconsistent. For example, it is impossible that Torn alone won one battle, but Torn and Andzhey together lost in another battle. The reverse is also true: if Telle can beat Torn and Andzhey together, she can beat Torn and Andzhey separately. The abilities of characters are not changing with time, thus the outcome of any battle is uniquely determined by the set of Gopniks fighting in that battle.

Formally, if a set U of Gopniks wins a certain battle against Telle, each battle between the girl and a superset of U must also be a win for Gopniks. Conversely, if a set V of Gopniks loses a certain battle, each battle between the girl and a subset of V must be a win for Telle.

You have to help Lukyanenko to fix all contradictions as soon as possible. It is not a very good idea to change the story very much, so you have to change the outcomes of the minimal possible number of battles.

Input

The first line contains integers N and K ($1 \le N, K \le 1000$) which are the number of Gopniks and the number of battles. Each of the next K lines contains description of a particular battle. A battle description has the following form. First comes an integer h ($1 \le h \le N$). Then follow h different integers t_i ($1 \le t_i \le N$) which are the indices of Gopniks which attacked the girl. After them follows one word describing the outcome of the battle: it is "`Girl`" if the girl has won this battle or "`Gops`" otherwise.

Output

The first line must contain a single integer r, the minimal number of battles which should have their result changed. The next line must contain exactly r integers q_i ($1 \le q_i \le K$), the indices of battles that must have their result changed according to your solution. The indices can be printed in any order. The battles are numbered from 1 to K in the order they appear in the input. If there are multiple optimal solutions, print any one of them.

Examples

standard input	standard input
3 4 1 1 Girl 1 2 Gops 2 2 3 Girl 3 1 2 3 Gops	1 2
2 2 2 1 2 Girl 2 1 2 Gops	1 2

Problem D. Circle Kingdom

Input file:	`standard input`
Output file:	`standard input`
Time limit:	1 second
Memory limit:	512 mebibytes

A far away Kingdom is shaped as an infinite 2D plane. Because of a hard political crisis, queen Ksenia decided to gather her representatives to an urgent military council. The lands ruled by Ksenia are filled with defensive fortresses. Each of them looks like a wall of a circular shape. The walls of different fortresses have no common points since that would lead to unnecessary expenses of maintaining non-typical architecture. Some of the fortresses may be located inside the others for greater security.

Once the queen representatives receive an invitation to the military council, they immediately move out from their houses, accompanied by some number of guards. At each point where a path of a queen's representative goes from the exterior of the fortress to its interior or vice versa, the representative must pay a tribute for passing through the gates (due to the corruption). Each fortress has a fixed size of tribute per person, and a representative needs to pay full tribute for himself and each of his guards.

Ksenia wants to choose such a meeting point for the military council so that the total tribute paid by her representatives is minimal possible (because they will be reimbursed from the state treasury). Any place in the Kingdom except the borders of the fortresses can be set as the meeting point. In addition, Ksenia can reduce her expenses by sending an inspection to no more than K selected fortresses. A fortress under inspection has its tribute eliminated.

On her way to the military council, Ksenia spends nothing, and all her representatives always choose the cheapest route, taking into account which of the fortresses were selected for inspection.

Input

The first line of input contains three integers N, M and K ($2 \le N \le 35\,000, 1 \le M \le 35\,000, 0 \le K \le N$). Here, N is the number of fortresses located at the Kingdom, M is the number of queen's representatives invited to the military council and K is the maximum number of inspections that can be sent. Next N lines describe the fortresses. Each of these lines contains four integers x, y, R and C ($-10^6 \le x \le 10^6$, $-10^6 \le y \le 10^6$, $1 \le R \le 2 \cdot 10^6$, $1 \le C \le 10^5$). Here, (x, y) is the center point of the fortress, R is its radius and C is the tribute per person for passing through a gate of this fortress.

The following M lines describe Ksenia's representatives. Each of these lines contains three integers x, y and L ($-10^6 \le x \le 10^6$, $-10^6 \le y \le 10^6$, $1 \le L \le 10^5$). Here, (x, y) are the coordinates of the representative's house and L is the total number of people that are going to travel from that house (including the representative). It is guaranteed that no two circles representing the fortresses will share a point, the locations of the representatives' houses are distinct and none of them lie on a fortress wall.

Output

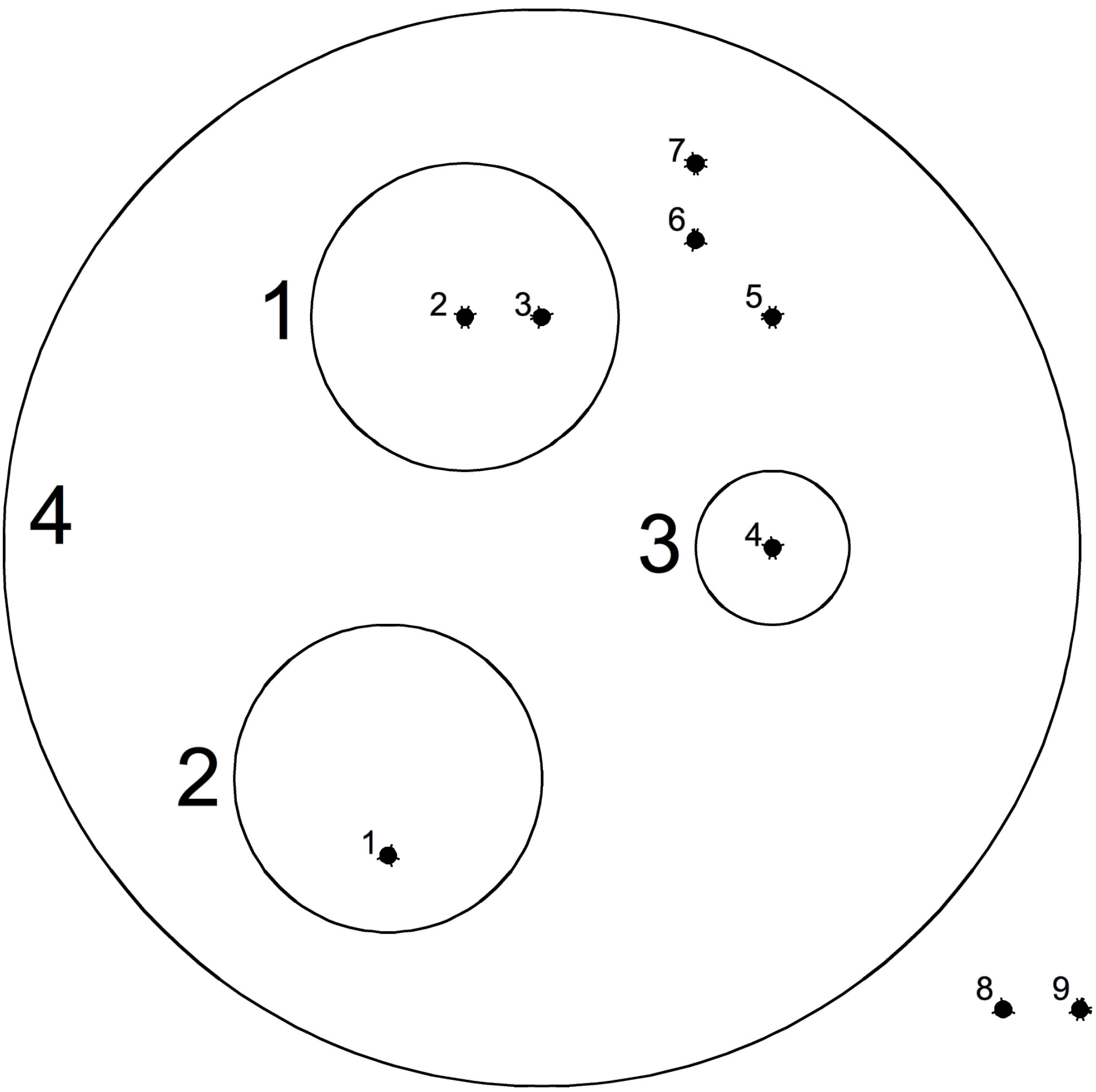
Output a single integer which is the minimal total tribute needed to gather all queen representatives at the military council.

Examples

standard input	standard input
4 9 1 6 10 2 1 5 4 2 1 10 7 1 200 7 7 7 1 5 3 10 6 10 1 7 10 1 10 7 1 10 10 1 9 11 1 9 12 1 13 1 1 14 1 1	12

Note

In order to achieve minimum expenses, queen Ksenia needs to send an inspection to the third fortress and choose the meeting point anywhere inside the second fortress.



Problem E. Game of Thrones

Input file:	*standard input*
Output file:	*standard input*
Time limit:	1 second
Memory limit:	512 mebibytes

Little Petya likes "Game of Thrones" series a lot! In fact, he likes it so much that he created his own setting of the game. In this setting, there are three kingdoms, a king and a queen. Both the king and the queen want to rule over all of the three kingdoms. The problem is, their forces are limited so the best they can do is conquer two of the kingdoms for a short period of time and collect the taxes. The king and the queen are playing the game by doing so. They take turns alternately, starting, of course, with the queen. On each move, a player collects taxes from any two kingdoms. Each kingdom also has a limited amount of resources known in advance, so for each kingdom, the taxes can be collected only a known limited number of times. The royal majesty who can't find two kingdoms to conquer and collect the taxes loses the game.

For this problem, you may assume that little Petya plays the role of the king and little Masha is the queen. Your task is to find out who will win the game, provided that both players play optimally.

Input

The first and only line of input contains three space-separated integers A_1, A_2 and A_3 ($0 \le A_i \le 10^6$) which are the numbers of times taxes can be collected from the first, second and third kingdoms respectively.

Output

Output "`Petya`" if Petya has a winning strategy if both players play optimally, and "`Masha`" otherwise.

Examples

standard input	standard input
`1 2 3`	`Masha`
`1 2 2`	`Petya`

Note

In the example, Masha makes the first move, conquering the 1st and the 3rd kingdom. After that, only the 2nd and the 3rd kingdoms can be conquered. Petya conquers them in his first move, and Masha conquers them afterwards, winning the game.

Problem F. Petya and Matches

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

Little Petya likes puzzles very much. Recently he received a puzzle as a gift from his mother. It can be represented as a table with N rows and M columns divided into $N \times M$ equal square-shaped cells. There are several matches located on the edges of some cells. The goal of the puzzle is to move as few matches as possible to achieve a configuration where every cell has a match on at least one of its edges. An edge of any cell can contain at most one match. Note that if two cells share an edge, there can be at most one match lying on it.

Input

The first line of input contains the number of test cases T. After that, T test cases follow. The sum of $N \cdot M$ over all test cases in a single input does not exceed 10 000.

The first line of each test case contains two integers N and M ($1 \leq N \leq 30$, $1 \leq M \leq 30$). Each of the next N lines contains M integers from the range $[0, 15]$. A number in the j-th position of the i-th row of the table encodes the information about the matches lying on the edges of the cell (i, j). The k-th bit in this number is 1 if and only if there is a match on the k-th edge of the cell. Bits are numbered from the least significant to the most significant one, starting from zero. The directions are numbered in the following order: up, right, down, left. It is guaranteed that the description of the table is consistent, that is, every match that lies on an edge belonging to two different cells appears in binary representations of both of them.

Output

For each test case, output one line containing the minimal number of matches that needs to be moved. If the goal can not be achieved, output -1 instead of that number.

Examples

standard input	standard input
4 1 1 3 2 2 2 8 0 0 2 2 15 8 1 0 2 3 15 8 0 1 0 0	0 -1 1 2

Problem G. Petya and Arrays

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

Little Petya likes arrays of integers very much. The only thing he likes more than arrays is playing with little Masha. It appears that she has just received three integers N, P and A as a gift from her mother. To test the strength of their friendship, Masha asked Petya to find the number of arrays consisting of N integers with the following properties:

- Every element is between 1 and P, inclusive.
- Number A is not present in the array.
- There is no continuous sub-array in which the sum of elements is divisible by P.

Help Petya to find the number of such arrays. This number can be very large, so output it modulo 1 000 000 007 ($10^9 + 7$) in order not to scare the children.

Input

The only line of input contains three integers N, P and A ($1 \le N \le 85$, $1 \le A \le P \le 1\,000\,000\,000$).

Output

Output the number of arrays Masha is asking for modulo $10^9 + 7$.

Examples

standard input	standard input
2 4 1	3
3 4 2	2
80 1000000000 1000000000	395358019

Problem H. Saw, Shura, Saw!

Input file:	*standard input*
Output file:	*standard input*
Time limit:	8 seconds
Memory limit:	512 mebibytes

Two famous rogue-losers Misha Panikovskiy and Shura Balaganov are playing a game. They are given a weighted rooted tree. On each move, a player cuts one of the tree's edges with the largest cost. When the edge has been cut, the whole subtree which is no longer connected with the root falls down and doesn't take part in the game anymore. The rogue-loser who makes the last cut becomes a rogue-winner and gets a golden weight hidden under the root of the tree.

Your task is to determine the name of the winner provided that both rogue-losers use the optimal strategy. Players make their moves alternately, Misha makes the first move.

Input

The first line of input contains the amount n ($2 \le n \le 10^6$) of vertices in the tree. The following $n-1$ lines contain triples of integers a_i, b_i and c_i ($1 \le a_i, b_i \le n$, $1 \le c_i \le 10^6$) which are the numbers of the vertices connected by the i-th edge and its cost. The root is the vertex with number 1.

Output

If Misha Panikovskiy can guarantee the victory to himself, output "`Misha`", otherwise output "`Shura`".

Examples

standard input	standard input
3 1 2 5 2 3 10	Shura
5 1 4 2 3 5 1 2 5 2 5 1 1	Misha

Problem I. All Along The Watchtower

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

Little Petya likes Bob Dylan very much. In fact, so much that he wants to take a closer look at the situation described in the famous song "All Along the Watchtower". In his imagination, there are exactly three watchtowers located in the corners of an isosceles right triangle on the plane at $(0, 0)$, $(20\,000, 0)$, $(0, 20\,000)$. There are N princes, numbered from 1 to N. For Petya's purposes, we can treat them as points on the plane in the interior of the triangle. From each watchtower, all princes can be seen, and the indices of princes were recorded in counterclockwise order as seen from this watchtower.

The song was written a while ago, so the princes are long gone, but the records in each watchtower were kept. Now Petya wonders if the records are correct and there indeed exist N points inside the triangle such that they are seen from each watchtower exactly as given in the records. Help Petya answer that question, and if such set of points exists, output one of the possible sets.

Input

The first line contains one integer N $(3 \le N \le 6)$ which is the number of princes.

The next line contains N space-separated integers which are a permutation of numbers 1 through N, the order in which princes were seen from the first tower, located at $(0, 0)$, in counterclockwise order.

The next two lines of input describe orders in which princes were seen from the second and the third towers (located at $(20\,000, 0)$ and $(0, 20\,000)$ respectively) in the same format.

Output

If the records contains mistakes and there are no possible placements of princes inside the triangle that satisfy the records, print "`NO`" on the first line of the output (quotes for clarity only).

If there exists a placement that satisfies the records, print "`YES`" without quotes on the first line of the output. Each of the next N lines should contain two space-separated real values. The $(i+1)$-st line should contain coordinates of the i-th prince in your placement.

The following conditions must be satisfied:

1. No prince should be located closer than 10^{-6} to any side of the triangle.
2. The distance between any prince and any line that goes through one of the towers and some other prince should be no less than 10^{-6}.

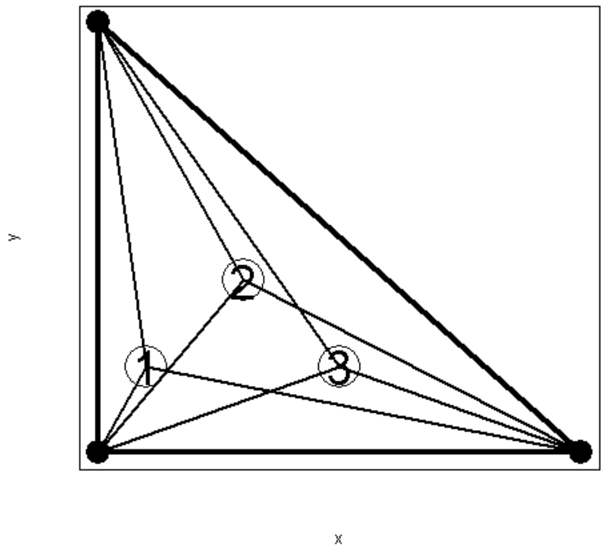
It is guaranteed that, if some valid placement of princes inside of the triangle exists, then there exists a placement that satisfies the conditions above.

Examples

standard input	standard input
3 3 2 1 2 3 1 1 2 3	YES 1 2 3 4 5 2
3 1 2 3 1 2 3 1 2 3	NO

Note

The angles are measured in counter-clockwise direction from the positive direction of X axis. Therefore, the second tower is seen from the first tower at the angle of 0 degrees, the third tower is seen from the first tower at the angle of 90 degrees, and the angle from each tower to each prince lies in $[0; 360)$. Note that princes must not necessarily be located at integer coordinates.



Problem J. Yet Another Binary Tree Problem

Input file:	*standard input*
Output file:	*standard input*
Time limit:	3 seconds
Memory limit:	512 mebibytes

Ksusha and Sasha work as senior BGEs[1] at a major outsourcing company. They have developed yet another board game called LRSRBTg[2]. Now they are going to play LRSRBTg the very last time before sending it to their customers.

The board for LRSRBTg consists of a single perfect rooted binary tree with bonuses located in some of the $N = 2^h - 1$ vertices. Note that there can be at most one bonus located in a single vertex. In addition to the board, there are infinitely many black and white pieces included to the LRSRBTg package. The rules that have been invented by Ksusha and Sasha are pretty simple: initially there is only one white piece located in some vertex V_{start} of the tree and two players alternate their turns during the game. At their turns players can perform one of the following three moves:

1. "Go left" — move all the pieces remained on the board (regardless of their color) to the left descendants of the vertices they are currently located in.
2. "Go right" — move all the pieces remained on the board (regardless of their color) to the right descendants of the vertices they are currently located in.
3. "Split" move consists of three sub-steps:
   - Replace each of the pieces remained on the board (regardless of color) with 2 pieces: one of them being black and the other being white.
   - Apply "Go left" move to all white pieces and only them.
   - Apply "Go right" move to all black pieces and only them.

As you can notice these moves sometimes can lead to the situation when some pieces are falling off the board. If the "Go left(right)" move is applied to piece which located in vertex that does not have left(right) descendant then that piece will fell off the board. In particular this implies that any of the moves 1-3 will lead to falling off the board pieces that were located in the leaves prior to the move.

Eventually, when there are no pieces left on the board the game ends. To determine the winner players count the total number of bonuses that are located at the vertices visited by at least one of the pieces. If the number of bonuses is even then Ksusha wins the game, otherwise Sasha wins. Given the tree structure and the location of all bonuses determine who will be the winner assuming that both Ksusha and Sasha play optimally. Ksusha makes the first move.

Input

The first line contains an integer $N = 2^h - 1$ ($1 \le N \le 32\,767$), the number of vertices. The next N lines describe a perfect rooted binary tree used for LRSRBTg. In the i-th of these lines three numbers are listed: $left$, $right$ and $bonus$ where:

- $left$ is the number of left child of i-th vertex or 0 if it doesn't have a left child.
- $right$ is the number of right child of i-th vertex or 0 if it doesn't have a right child.
- $bonus$ is equal to 1 if the i-th vertex contains a bonus and 0 otherwise.

Output

In the $i - th$ line output the winner of the game ("`Ksusha`" or "`Sasha`", quotes for clarity only), assuming that the starting position is located in the $i - th$ vertex (playing the game with $V_{\text{start}} = i$) and Ksusha plays first.

Examples

standard input	standard input
3 2 3 0 0 0 1 0 0 0	Ksusha Sasha Ksusha

Note

[1] Board Game Engineers, almost a real job title

[2] Abbreviation from Left-Right-Split Rooted Binary Tree game

A perfect binary tree is a binary tree with $2^h - 1$ vertices where h is the maximal depth of the tree, and each internal vertex has two descendants.