

Problem A. Lucky pairs (Junior)

Input file: `stdin`
Output file: `stdout`
Time limit: 1 s
Memory limit: 256 Mb

Ksenia is a proud owner of an array a_i length N , where each of the numbers $1, 2, \dots, N$ is present exactly once. Ksenia considers a pair of indices (i, j) lucky if $a_i < a_j$ and $i > j$ at the same time. Find the number of lucky pairs of indices in a given array.

Input

The first line contains the only number — N ($1 \leq N \leq 1000$). The second line contains N numbers separated by a space, which represent the array a_i .

Output

Print the number of lucky pairs of indices in a given array.

Examples

stdin	stdout
3 3 2 1	3
5 3 1 4 5 2	4

Problem B. Laser (Junior)

Input file: `stdin`
Output file: `stdout`
Time limit: 1 s
Memory limit: 256 Mб

Ksenia is the most responsible worker in the Research Institute. So she was asked to make a very important experiment: to melt the chocolate bar with a new laser device. The device consists of a rectangular field of $n \times m$ cells and a robotic arm. Each cell of the field is a 1×1 square. The robotic arm has two lasers pointed at the field perpendicularly to its surface. At any one time lasers are pointed at the centers of some two cells. Since the lasers are on the robotic hand, their movements are synchronized: if you move one of the lasers by a vector, another one moves by the same vector.

The following facts about the experiment are known:

- initially the whole field is covered with a chocolate bar of the size $n \times m$, both lasers are located above the field and are active;
- the chocolate melts within one cell of the field at which the laser is pointed;
- all moves of the robotic arm should be parallel to the sides of the field, after each move the lasers should be pointed at the centers of some two cells;
- at any one time both lasers should be pointed at the field. Ksenia doesn't want to become a second Gordon Freeman.

You are given n , m and the cells (x_1, y_1) and (x_2, y_2) , where the lasers are initially pointed at (x_i is a column number, y_i is a row number). Rows are numbered from 1 to m from top to bottom and columns are numbered from 1 to n from left to right. You are to find the amount of cells of the field on which the chocolate can't be melted in the given conditions.

Input

The first line contains one integer number t ($1 \leq t \leq 10000$) — the number of test sets. Each of the following t lines describes one test set. Each line contains integer numbers n , m , x_1 , y_1 , x_2 , y_2 , separated by a space ($2 \leq n, m \leq 10^9$, $1 \leq x_1, x_2 \leq n$, $1 \leq y_1, y_2 \leq m$). Cells (x_1, y_1) and (x_2, y_2) are distinct.

Output

Each of the t lines of the output should contain the answer to the corresponding input test set.

Examples

stdin	stdout
2 4 4 1 1 3 3 4 3 1 1 2 2	8 2

Problem C. Christmas gifts (Junior)

Input file: `stdin`
Output file: `stdout`
Time limit: 1 s
Memory limit: 256 Mb

Ksenia prepares Christmas gifts for parents. This year she wants to present them a set of Christmas balls. Ksenia doesn't want to present a set of balls which was created by professional designers, because the best gift is the one that she makes (or collects) by herself. Therefore, Ksenia decided to put together a set of Christmas balls in the store «Garlands and tourism».

The store «Garlands and tourism», sells **numbered** balls painted in one of the N colors. It is known that all Christmas balls' numbers are **different** and the amount of Christmas balls of each color is equal to K items. Ksenia wants to buy a set of exactly N balls in such way that she will have exactly one ball of each color.

How many different sets of N balls can Ksenia present to parents? The answer should be given in modulo M .

Input

The first line contains the number of test cases T ($1 \leq T \leq 1000$). Each of the next T lines contains three numbers: N , K and M ($1 \leq N, K, M \leq 10^{18}$).

Output

For each test case, output the remainder of division by M of the number of different options to choose N balls from given $N \cdot K$ so that they have different colors. Note that all the balls are unique due to number on them.

Examples

stdin	stdout
2	32
5 2 1000	25
2 5 1000	

Problem D. Deputies on the tree (Junior)

Input file: **stdin**
Output file: **stdout**
Time limit: **1 s**
Memory limit: **256 Mb**

In consequence of excessive decentralization the capital has been eliminated in country X. Now country X has a complicated political situation, which can only be resolved after gathering the new council of deputies.

Deputies live in each of the N cities. City i ($1 \leq i \leq N$) has exactly d_i deputies living in it. There are $N - 1$ two-way roads between the cities of country X that provide exactly one way to get from any city to any other one. For each road i ($1 \leq i \leq N - 1$) we know the cities u_i and v_i it connects and the cost w_i of travelling by this road for a single person.

Since country X experiences serious problems with finances. To save the country deputies must meet in some city, so that the total cost spent on travelling will be the lowest possible. To simplify the problem, you as the acting Prime Minister of X are permitted to declare a free passage for $0 \leq K \leq N - 1$ roads.

Input

The first line contains two numbers N, K ($1 \leq N \leq 1000$, $0 \leq K \leq N - 1$). The second line contains N integers $1 \leq d_i \leq 100000$. The i -th of them is the number of deputies living in the i -th city. The remaining $N - 1$ lines describe the roads of country X. Each road is defined in a separate line by three numbers: u_i, v_i, w_i ($1 \leq u_i, v_i \leq N$, $1 \leq w_i \leq 100000$). It is guaranteed that a deputy can reach any city from any other one by specified roads.

Output

Print the minimum amount of money required to gather all deputies in one city.

Examples

stdin	stdout
5 3 8 5 1 8 7 1 3 8 2 3 9 3 5 7 3 4 8	45

Problem E. LinearMapReduce

Input file: `stdin`
Output file: `stdout`
Time limit: 1 s
Memory limit: 256 Mb

In computing paradigm linearMapReduce the basic units of data are records. Every record consists of two fields — the key and the value. Key and value in linearMapReduce are integers in the range $[0, 10^9 + 6]$. Data processing paradigm linearMapReduce uses operators of two types: linearMap and much less popular linearReduce. Operator linearMap converts input key-value pair $(Key_{in}, Value_{in})$ into another key-value pair $(Key_{out}, Value_{out})$ according to the following rule:

$$Key_{out} = (a_{11} \cdot Key_{in} + a_{12} \cdot Value_{in}) \bmod (10^9 + 7)$$

$$Value_{out} = (a_{21} \cdot Key_{in} + a_{22} \cdot Value_{in}) \bmod (10^9 + 7)$$

Thus, the operator linearMap can be completely determined by 4 integers: $a_{11}, a_{12}, a_{21}, a_{22}$.

Ksenia supports project X where computational paradigm linearMapReduce is widely used. During the next scaling Ksenia needs to effectively solve the following problem:

N linearMap operators $M_1, M_2, \dots, M_N$ are described by their coefficients. There are incoming queries consisting of four numbers: $Key, Value, a, b$, which are the key-value pair and two indices in the operator sequence. In response to a query output the result of applying operators with indices from the a -th to the b -th to record $(Key, Value)$.

Note that in case $a \leq b$ we have to apply operators in the following order: $M_a, M_{a+1}, \dots, M_{b-1}, M_b$, and in case $a > b$ the order should be $M_a, M_{a-1}, \dots, M_{b+1}, M_b$.

Input

The first line contains a single integer $1 \leq N \leq 100000$ the number of linearMap operators. Each of the following N lines contains linearMap operators described one per line. Each operator is described by four integers separated by spaces: $a_{11}, a_{12}, a_{21}, a_{22}$, where $0 \leq a_{11}, a_{12}, a_{21}, a_{22} \leq 10^9 + 6$. The $N + 2$ -th row contains a single integer Q ($1 \leq Q \leq 100000$), the number of queries. Every query consists of 4 integers: $Key, Value, a, b$ separated by a space, where $0 \leq Key, Value \leq 10^9 + 6$ and $1 \leq a, b \leq N$.

Output

For each query output the key-value pair obtained after applying operators from the a -th to the b -th one.

Examples

stdin	stdout
3 2	15 15
1 0 0 1	20 20
1 2 2 1	
1 3 3 1	
5 5 1 2	
5 5 3 3	

Problem F. Circles and trees

Input file: `stdin`
Output file: `stdout`
Time limit: `1 s`
Memory limit: `256 Mb`

Specifies the root directed tree consisting of N vertices, numbered from 1 to N . You need to place N circles x_i, y_i, r_i on the plane so that:

1. Each of the numbers x_i, y_i, r_i — integer, $-10^6 \leq x_i, y_i \leq 10^6$, $1 \leq r_i \leq 2 \cdot 10^6$;
2. Circles have no common points;
3. Circle i is embedded in the circle j only if the vertex j of a given rooted tree can get to vertex i .

Input

The first line contains the number N ($1 \leq N \leq 20000$) — number of vertices of the tree. The next N lines is a description of the root oriented tree. In the row i contains exactly one number p_i - ancestor of vertex $i - 1$ if $i - 1$ is not a root and -1 otherwise.

Output

The first line should display the number N . In the next N rows you should print a circle satisfying 1-3 in the format « $x_i y_i r_i$ » (without quotes).

Examples

<code>stdin</code>	<code>stdout</code>
3	4 0 4
-1	2 0 1
1	5 0 1
1	

Problem G. Petya and the game

Input file: **stdin**
Output file: **stdout**
Time limit: **1 s**
Memory limit: **256 Mb**

Little Petya loves playing computer games. His mother has recently gave him a game where he has to rule a kingdom. The kingdom consists of N cities numbered from 1 to N , where the capital has number 1. Some cities are connected by two-way roads with length given in kilometers. There are $N - 1$ roads in total and is guaranteed that one can get from any town to the capital in only one way without going by the same road more than once. Every city of the kingdom, except of the capital, has exactly one harbinger living in it. Harbingers are characterized by their speed and the time they spend for preparations before the trip. For different harbingers these characteristics may vary. When the city is attacked a harbinger is sent to the capital. He is preparing for the trip and then moves one road towards the capital. Then, if he has not yet reached the capital, he can either continue moving to the capital (without spending additional time on preparation) or pass the message to the harbinger located in the city where he stays right now. In this case, everything starts over again: the harbinger is preparing and then leaves for the capital performing exactly the same actions. Thus, any number of harbingers can participate in delivering the message to the capital.

Help Petya determine the minimum time that it takes for the message to be delivered from each city to the capital.

Input

The first line contains a single integer N ($1 \leq N \leq 100\,000$). Each of the following $N - 1$ lines contain three integers u , v and d ($1 \leq u \leq N$, $1 \leq v \leq N$, $1 \leq d \leq 10\,000$), which are the numbers of cities connected by a road and its length in kilometers. Then there are $N - 1$ lines. The i -th of them contains two numbers S_i and V_i ($0 \leq S_i \leq 10^9$, $1 \leq V_i \leq 10^9$) characterizing the harbinger from the city $i + 1$. S_i is the time in minutes that a harbinger needs to get ready for a trip and V_i is the time in minutes that a harbinger spends to pass 1 kilometer of a road.

Output

Print the $N - 1$ numbers separated by spaces, where the i -th number has to be equal to the smallest time in minutes of message delivery from the $i + 1$ -th city to the capital.

Examples

stdin	stdout
5 1 2 20 2 3 12 2 4 1 4 5 3 26 9 1 10 500 2 2 30	206 321 542 328

Problem H. Petya and arrays

Input file: `stdin`
Output file: `stdout`
Time limit: `1 s`
Memory limit: `256 Mb`

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 two integers N and P 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.
- 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 first line of input contains the number of test cases T ($1 \leq T \leq 2500$). Each of the next T lines contains two integers N and P ($1 \leq N \leq 50$, $1 \leq P \leq 50$).

Output

For each test case output the number of arrays Masha is asking for modulo $10^9 + 7$. Separate answers for different test cases by a newline.

Examples

<code>stdin</code>	<code>stdout</code>
3	0
5 4	2
2 3	6
2 4	

Problem I. Petya and arrays 2

Input file: `stdin`
Output file: `stdout`
Time limit: `1 s`
Memory limit: `256 Mb`

Little Petya loves arrays of integers. Recently he has received an array consisting of N integers as a gift from his mother. Petya decided to find there a segment of consecutive elements for which the value of the expression $A \cdot S + B$ is maximized. A and B are given, and S is the sum of all elements from the selected segment. Please note that the selected segment must have non-zero length.

Input

The first line contains 3 integers N ($1 \leq N \leq 10^6$), A and B ($-10^6 \leq A \leq 10^6$, $-10^6 \leq B \leq 10^6$).

Output

Output the maximum value of the expression $A \cdot S + B$.

Examples

stdin	stdout
4 2 3 0 -1 -1 0	3
5 3 3 5 -1 4 5 2	48

Problem J. Petya and rectangles

Input file: `stdin`
Output file: `stdout`
Time limit: 1 s
Memory limit: 256 Mb

Little Petya likes rectangles a lot. Petya gave his mother a list of rectangles he wants to get for Christmas. Each rectangle has width w and height h . Petya's mother wants to make Petya happy and buy all the rectangles from his list. She went to the store and found out that the price of a rectangle is equal to its area. Luckily, the store has Christmas sale, so you can buy rectangles in sets. The cost of one set is calculated as the width of the widest rectangle multiplied by the height of the highest rectangle of this set. Note that you can't rotate the rectangles (thus swap their width and height). Help Petya's mother buy all the rectangles from the Petya's list spending a minimum amount of money on this.

Input

First line contains an integer N ($1 \leq N \leq 100\,000$), which is the number of rectangles in the Petya's list. Each of the next N lines contains 2 positive integers that do not exceed 10^6 , which represent width and height of the rectangle.

Output

Output the smallest amount of money that Petya's mom can spend to buy all the rectangles from the Petya's list.

Examples

stdin	stdout
4 100 1 15 15 20 5 1 100	500
5 1 10 2 20 3 30 4 40 10 1	170

Problem K. Yet Another Binary Tree Problem

Input file: `stdin`
Output file: `stdout`
Time limit: 1 s
Memory limit: 512 Mb

Ksusha and Sasha work as senior BGEs at a major outsourcing company. They have developed yet another board game called LRSRBTg. 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 $N = 2^h - 1$ ($1 \leq N \leq 32767$), 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

stdin	stdout
3	Ksusha
2 3 0	Sasha
0 0 1	Ksusha
0 0 0	

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.