Problem A. Absorbing The Point

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

Given the sequence S of N points (x, y) and the point P.

Your task is to answer the following queries: given two indices l and r, defining the subsequence $S_l, \ldots, S_r$ as the initial set. You may several times do the following process: choose two points from the current set and add to the set arbitrary point from the segment connecting chosen points. Check if you can add the point P in set by this way.

Input

The first line of the input contains three integers N, X_p and Y_p — the number of points in the sequence S and coordinates of the point P, respectively ($1 \leq N \leq 3 \cdot 10^5$, $-10^9 \leq X_p, Y_p \leq 10^9$).

i-th of the following N lines contains the coordinates of the point S_i — two integers X_i and Y_i ($-10^9 \leq X_i, Y_i \leq 10^9$).

Then one integer Q follows — the number of the queries ($1 \leq Q \leq 3 \cdot 10^5$).

Each of the following Q lines contains two integers l and r ($1 \leq l \leq r \leq N$) — starting and ending indices of the contiguous subsequence that is used as the initial set.

Output

For each query print "`Yes`" if the point P can be added to the set using the process described in the statement, or "`No`" otherwise.

Examples

standard input	standard output

5 1 2
0 0
0 3
3 3
3 0
0 0
4
1 3
3 5
2 4
1 5
	Yes
	No
	Yes
	Yes

Problem B. Black and White Map

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

Given a closed polyline on the infinite plane; it is known that any point of the plane lies on no more than two edges of the polyline. A polyline divides the plane to the *regions*; border of regions is formed by the polyline's edges and its parts; an exterior of the polyline is considered an infinite region. We say that two regions are *neighbors*, if there exists a segment of positive length that belongs to the border of both regions.

Now we want to choose some regions and paint them black, in such a way that for any two neighbors exactly one of them is black. Calculate the minimum possible total area of the black regions.

Input

First line of the input contains one integer n — number of the edges of the polyline ($3 \le n \le 500$).

Each of the next n lines contain two integers x_i and y_i — coordinates of one vertice of polyline. Vertices with coordinates (x_i, y_i) and (x_{i+1}, y_{i+1}) are connected by the edge, same as vertices (x_1, y_1) and (x_n, y_n), $-1000 \le x_i, y_i \le 1000$).

Output

Print miniumum total area of black regions with absolute error 10^{-4} or better. If it is impossible to build such a coloring, print -1 instead,

Examples

standard input	standard output

3
0 0
1 1
1 0
	0.5000

4
0 0
2 0
2 6
4 4
	6.0000

Problem C. Covering

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

Lets define a *route* in directed graph from vertice v_0 to vertice v_k as the sequence of vertices and edges

$$v_0,\ e_1,\ v_1, \dots, v_{k-1},\ e_k,\ v_k$$

where edge e_i connects vertices v_i and v_{i-1}. *Cost of the vertex* is defined as the number of outgoing edges from this vertex, and *cost of the route* is defined as the sum of costs of all vertices in this route.

Consider directed graph with n vertices and m edges with next property: for any edge e there exists the route from vertex 1 to vertex n that contains this edge. We define the *covering* as a set of routes from vertex 1 to vertex n, such that each edge is a part of at least one of these routes. Total cost of the covering is defined as the sum of costs of all routes in the covering.

Your task is to find the covering of minimum total cost.

Input

First line of the input file contains two integers n and m $(2 \le n \le 200,\ 1 \le m \le 1000)$ — number of vertices and edges of the given graph, respectively. Each of next m lines contains description of one edge — starting vertice a_i and ending vertice b_i. Note that there may be loops and multiple edges $(1 \le a_i, b_i \le n)$.

Output

Print the covering with minimum total cost in the next form: in first line print the cost, in second line print number of routes l, in next l lines print the routes, one per line. For each route first print integer l — number of vertices in the route, then l integers — vertices in the route, listed in order defined by this route.

Examples

standard input	standard output

4 4
1 2
2 3
3 2
3 4
	7
	1
	6 1 2 3 2 3 4

4 7
1 2
1 4
2 3
2 3
3 2
3 3
3 4
	17
	2
	7 1 2 3 2 3 3 4
	2 1 4

Problem D. Discover And Transform

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

This is an interactive problem.

The jury program got the string of length N, composed of characters '`A`', '`C`', '`G`' and '`U`'.

You know the length of the string and may ask the queries about the string. The queries have the form `?` L, R, C, where L and R are the starting and the ending index of the substring, and C – one of 4 characters mentioned above. As the answer you receive the number of the characters C in the substring starting from L-th character and ending at R-th character.

Your task is to transform string in the way such there are no more than $N/2$ of the characters of each type. To do that, you can do the following final action **once**, after all queries: choose the substring and replace in that substring all '`A`' with '`U`' and vice versa, and all '`C`' with '`G`' and vice versa.

Interaction Protocol

At the beginning, the jury program prints the integer N ($1 \le N \le 10^5$). Then you may ask no more than 36 queries in the form `?` L R C, where $1 \le L \le R \le N$, C — one of the characters '`A`', '`C`', '`G`' and '`U`'.

After those queries you shall transform the string with a command `!` L R ($1 \le L \le R \le N$). This command replaces all '`A`' with '`U`', all '`U`' with '`A`', all '`C`' with '`G`', all '`G`' with '`C`' to convert the given string to a string where are no letter with frequency strictly more than $N/2$.

If it is impossible to transform the given string using the action that is mentioned in the statement, print the string "`! -1 -1`" instead of the transformation command.

Don't forget to enter the end-of-line character and flush the output after each query and the final command.

Problem E. Exploring the Maze

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

Dora is walking into the maze and collecting the values. In this case maze is a tree, i.e. a connected graph that contains N vertices and $N-1$ edges. She wants to get maximum values by walking from a vertex A and stopping at a vertex B.

Dora is allowed to choose a vertex A (the value of A can not be 0) and a vertex B by herself. Walking from A and stopping at B, she must collect all the values on the road. Each vertex has a value. Dora tries to get values as large as she can. But she can never go back to any vertex she has passed.

However, there is a special way to calculate total values. Let's assume that Dora has passed M vertices from A to B. During the travel, Dora has successively collected M values worths W_i $(0 \le i < M)$. Vertex A has a value worth W_{M-1}. The next vertex on the road has a value worth W_{M-2}, ... At last, vertex B has a value worth W_0.

Given an integer P, the total value Dora will collect is calculated by the formula

$$MAX = \sum_{i=0}^{M-1} (W_i \times P^i).$$

It is guaranteed that W_i $(0 \le i < M)$ are less than P. The vertex A and B you choose can be same. But the value of A can not be 0. Output MAX modulo $(10^9 + 7)$.

Note that you need to make sure MAX as large as possible but NOT make sure the remainder as large as possible. And then, output value of each vertex (stating from vertex A) on the road in the best case.

Input

The first line contains an integer $T (1 \le T \le 200)$, indicating the number of test cases.

For each test case, the first and second line contain two integers N $(1 \le N \le 10^4)$ and P $(2 \le P \le 10^9)$, indicating the number of vertexes and the integer P.

Each of the following $N-1$ lines contains two integers a and b $(1 \le a, b \le N, a \ne b)$, indicating that there is an edge connecting vertex a and vertex b.

The following line contains N integers W_i $(0 \le W_i < P, \sum W_i > 0)$, the value of each vertex. It is guaranteed that at least one of W_i not equal to zero.

You can assume that sum of all N in the test cases does not exceed 1.3×10^6.

Output

For each test case, print two lines. First must contain the maximum value of treasures Dora can collect modulo $(10^9 + 7)$.

In the second line print the values of each vertex from vertex A to vertex B.

Example

standard input	standard output

2
8
2
1 2
2 3
3 4
4 5
2 6
6 7
7 8
1 0 0 0 0 0 0 0
9
923456789
1 2
2 3
1 4
4 5
1 6
6 7
1 8
8 9
1 2 3 2 0 2 0 2 3
	16
	1 0 0 0 0
	90485756
	3 2 1 2 3

Problem F. Favorite Restaurants

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

In the Bytesdorf village there are N restaurants numbered with sequential integers between 1 and N. The owners of the restaurants have their favorite restaurants they like to visit: if we will ask the owner of some restaurant for a recommendation, he will, besides his restaurant, recommend:

- his favorite restaurants;
- all favorite restaurants of owners of those restaurants
- and all favorite restaurants of owners of restaurants from previous step
- ...and so on...

Alice plans to visit several restaurants in the following way:

- The first restaurant will be chosen arbitrarily.
- Each next restaurant will be chosen from list of restaurants, recommended by owner of current restaurant, which was not visited by Alice before.
- Alice may decide to stop her restaurant trip at any time

Each restaurant have two prices for the main menu: X_i and Y_i. When the guest enters the restaurant, the owner asks who recommended the restaurant. If this person is in the recommendation list of owner of this restaurant, then guest pays X_i, otherwise (including the case when it is starting restaurant and there was no recommendation) guest pays Y_i.

Let M is the maximum number of restaurants that Alice can visit in this way. For each integer K between 1 and M you need to calculate minimum amount of money Alice need to visit exactly K restaurants.

Input

First line of the input contains an integer N ($1 \le N \le 1000$) — the number of restaurants. Each of following N lines begins with two integers X_i, Y_i — prices for two types of guests ($1 \le X_i, Y_i \le 10^4$), then the integer F_i follows — number of favorite restaurants for owner of i-th restaurant, then F_i integers are listed — numbers of the restaurants from favorite list. It is guaranteed that all those numbers are pairwise distinct and not equal to i.

Output

Print K lines (where K is maximum number of restaurants Alice can visit in a way, described in problem statement), j-th of those lines must contain one integer — minimum amount of money Alice need to pay to visit exactly j restaurants.

Example

standard input	standard output

4
100 200 1 2
200 300 1 3
200 250 2 2 4
200 300 0
	200
	450
	650
	950

9
100 100 0
300 400 1 4
350 500 1 2
550 600 3 7 3 2
900 300 2 7 6
250 400 1 5
900 900 2 9 8
400 500 1 9
500 400 0
	100
	550
	950
	1450
	2150
	3050

Note

Consider the first sample. The full list or recommendations of the owner of restaurant 1 contains restaurant 2, then the favorite restaurants of the owner of restaurant 2 (i.e. restaurant 3), then the favorite restaurants of the owner of restaurant 3 (i.e. restaurant 4 is added), so he may recommend restaurants 2,3,4. The owner of restaurant 2 may recommend restaurants 3,4. The owner of restaurant 3 may recommend restaurants 2,4. The owner of restaurant 4 does not have any favorite restaurants.

So for $K = 1$ Alice must use the second price (no recommendations are given), so the minimal answer is 200 (restaurant 1). For $K = 2$ Alice goes to restaurant 2, pays 250, then she goes to restaurant 3. Since 3 is in the recommendation list for 2, the first price may be used, so total price is 250. For $K = 3$ Alice starts at restaurant 1 with second price, then goes to restaurant 3 (with second price — 1 is not in any other recommendation list), then goes to restaurant 2 with first price, so she pays $200+250+200=650$. For $K = 4$ Alice continues to the restaurant 4, that have empty recommendation list, so she pays 300 and the answer for that K is 950.

Problem G. Game For One

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

Consider the following computer game for one player: given the permutation of length N.

There are events of 2 types: at the event of type 1 the player is given the range from l-th to r-th (inclusive) positions and may reorder those elements as she want to; at the event of type 2 the computer choose the integer at the i-th position. If this integer is equal to 1, the player is awarded with one bonus point.

You are given list of events in the game. Calculate the maximal number of bonus points that can be collected by the player.

Input

The first line of the input contains one integer N — the length of the permutatuion ($1 \le N \le 3 \cdot 10^5$).

On the next line, the permutation of the length N follows (N pairwise distinct integers between 1 and N, inclusively).

Then one integer Q follows — the number of events ($1 \le Q \le 3 \cdot 10^5$).

The following Q lines contains the queries, in order, one query per line. Each query is described by three integers and starts with the type t ($1 \le t \le 2$). If $t = 1$, then two integers follow, l and r ($1 \le l \le r \le N$) — the leftmost and rightmost position of the range, where the player may reorder the elements. If $t = 2$, then second integer i is the index of the element chosen by computer ($1 \le i \le N$), the third integer then is equal to 0.

Output

Print one integer — the maximum amount of bonus points the player can get while reordering the elements at the type 1 queries optimally.

Examples

standard input	standard output

6
3 1 4 6 5 2
4
1 2 4
2 5 0
1 4 6
2 5 0
	1

4
4 3 2 1
4
1 1 4
2 1 0
2 1 0
2 3 0
	2

Problem H. Horizontals and Verticals

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

Given the grid $N \times M$. Each cell of the grid is painted in one of two colors: black or white.

In one turn is allowed to choose a cell and reverse colors on the same horizontal and the same vertical with selected cell (including this cell), i.e. black cells are replaced with white ones and vice versa.

Given initial coloring check if it is possible to make all cells on the grid white.

Input

First line of the input contains two space-separated integers N and M ($1 \leq N, M \leq 50$).

Then N lines follow — description of the initial position. Each line contains a string of length M, consisting of letters 'B' if initially corresponging cell is black and 'W' otherwise.

Output

If it is impossible to paint all cells white, print "No". Otherwise in first line print "Yes", in second — number of turns, each of next lines must contain two integers — row and column of cell, selected at this turn.

Examples

standard input	standard output
2 3 WWB BBB	Yes 1 2 3
3 3 WWW WBW WWW	No

Problem I. Incident Vertices

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

Given two trees, each tree has N vertices, enumerated with 1. Your task is to answer Q queries. Each query contains four indices A, B, C and D.

The answer on the query is $\sum_{i=1}^{n} f(i)$, where $f(i)$ is defined by the following way:

- $f(i) = 1$ if the vertex with the index i is placed on the path from the vertex A to the vertex B in the first tree **and** the vertex with the index i is placed on the path from the vertex C to the vertex D in the second tree,
- $f(i) = 0$, otherwise.

Input

The first line of the input contains three integers N, Q and P — the number of the vertices of the tree, the number of the queries and the parameter, respectively ($1 \le N \le 2 \cdot 10^5$, $1 \le Q \le 2 \cdot 10^5$, $0 \le P \le 1$).

Then $N - 1$ pairs of integers v, u follow — the edges of the first tree ($1 \le u, v \le N$). You may assume that the given graph is tree.

Then the description of the second tree is listed in the similar format.

Then the information of Q queries is given. Each query contains four integers a_i, b_i, c_i and d_i ($1 \le A_i, B_i, C_i, D_i \le N$). The actual values of A_i, B_i, C_i and D_i are calculated by the formula $A_i = (a_i + answer_{i-1} \cdot P - 1) \bmod N + 1$, $B_i = (b_i + answer_{i-1} \cdot P - 1) \bmod N + 1$, $C_i = (c_i + answer_{i-1} \cdot P - 1) \bmod N + 1$, $D_i = (d_i + answer_{i-1} \cdot P - 1) \bmod N + 1$, where $answer_{i-1}$ is the answer on the previous query, $answer_0 = 0$.

Output

For each query, print a line with one integer — the answer to that query.

Examples

standard input	standard output

5 1 0
1 4
2 3
1 5
1 2
1 3
2 3
1 5
4 3
5 3 2 4
	2

Problem J. Journeys With Rent-a-car

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

There are N cities connected by M bidirectional roads. For each road, its length L is given.

In each of N cities works its own the rent-a-car service. The cars from the i-th city, being fully fueled, can cover the distance D_i.

To travel between cities, the citizen may rent a car in one city, drive to another, then either refuel the current car fully and continue the travel to the next city, or end the current rent and switch to the car from the new city. Cars may be refueled only in the cities.

For each city calculate the number of the cities that are reachable from this city using the way described above (with zero or more car changes). The city itself shall be counted too.

Input

The first line of the input contains two integers N and M ($1 \le N, M \le 2 \cdot 10^5$). The second line contains N integers D_i — the travel distance for the cars from i-th city, respectively ($1 \le D_i \le 10^9$).

Each of the following M lines contains description of one road and consists of three integers A, B and L — the cities connected by the road and the road length, respectively ($1 \le A, B \le N$, $1 \le L \le 10^9$).

Output

Print N integers. i-th of those integers shall denote the number of cities reachable from the city i.

Examples

standard input	standard output

5 7
7 7 7 7 7
1 2 3
1 3 8
2 5 7
4 5 10
1 5 6
2 3 9
3 5 8
	3 3 1 1 3

5 4
1 7 8 5 3
1 3 7
1 5 7
3 5 7
2 4 7
	1 2 3 1 1

5 5
5 9 6 1 3
1 5 7
1 2 4
2 5 9
4 3 2
3 5 6
	5 5 3 1 1

Problem K. Ksir

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

Kevin and Kate have played the game Risk a thousand times, so now they are trying to invent a new game called Ksir playing on the same board in the form of a geographic map. The board contains N countries enumerated by sequential integers from 1 to N, and it is known which pairs of countries are neighbors. Note that some countries may be neighbors even if they do not physically share the boundary.

Before the start of the game, a number of knights have been set up in each country, and some countries have declared as "attackers", while the remaining "defenders".

Game processed in turns. Player who cannot make a turn loses the game. Kate is first to move. At his turn player choose one of two types of actions:

1. Attack:
   - The player selects the attacker country A with T_A knights and neighborung defender country M with T_M knights.
   - Action can be done only when $T_M > 0$.
   - Then each knight from the country A destroys one knight from the country M.
   - At the end of the turn in country M will remain $T_M - T_A$ knights (or 0 in case when $T_A > T_M$).
2. Help:
   - The player selects two adjacent defender countries M and O with T_M and T_O knights, respectively.
   - Action can be done only when $T_M > 0$.
   - If the T_M is odd, the player adds a new knight to country M.
   - Then exactly half of the knights from country M is moved to country O.

Note that countries do not belong to individual players; each player in their turn can choose either which the two adjacent countries, provided that the move is allowed.

Since the board contains N countries, there is 2^N ways to set the countries to attacker and defender. For every possible distribution, Kate and Kevin will play one game. They are interested in how number of games won by Kate and number of games won by Kevin provided both players play optimally. In some cases no player can secure a win (for example if there are no attacker countries).

Input

First line of the input contains one integer N ($2 \le N \le 40$) — number of countries. Next line contains N integers T_i — number of knights in i-th country ($1 \le T_i \le 4 \cdot 10^4$).

Next line contains one integer M ($1 \le M \le 780$) — number of pairs of neighbors. Each of next M lines contain two distinct integers between 1 and N — numbers of countries who are a neighbors. Each pair of countries is listed at most once.

Output

In first line of the output print number of games won by Kate, in second — number of games won by Kevin.

Examples

standard input	standard output

2
100 100
1
1 2
	2
	1

5
7 5 3 4 5
5
1 2
2 3
3 1
3 4
3 5
	5
	8

Problem L. Lexicographically Maximal Sequence

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

Given the sequence of non-negative integers a_1, a_2, $\ldots$, a_n of length n. You may apply the following operation arbitrary number of times: choose an index i $(2 \le i \le n-1)$ and replace a_{i-1} with $a_{i-1} \oplus a_i$, and a_{i+1} with $a_{i+1} \oplus a_i$, where $\oplus$ is the bitwise XOR operation.

Print the lexicographically **maximal** sequence you can obtain from the given sequence.

Input

The first line of the input contains one integer N $(1 \le N \le 150\,000)$. The second line contains N integers A_i $(0 \le A_i \le 10^{18})$.

Output

Print N integers — the lexicorgraphically maximal sequence you can obtain using those operation several times.

Example

standard input	standard output
3 1 3 2	2 3 1
5 31 31 31 31 31	31 31 31 31 31

Problem M. Matches

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

Mary found a pile of chips and matches. The chips are arranged in a rectangular grid $R \times C$.

For two chips we say that they are adjacent if they are in the same or adjacent row, and in the same or adjacent column. Each chip can therefore have up to 8 neighbor chips.

Then Mary sets up matches between some adjacent chips. We say two chips are *connected* if there is a path that runs from the first to the second chip through one or more matches. Every pair of chips is connected and no two matches cross each other.

In the example below we see nine chips and matches that link them. In order to uniquely record match positions, for each chip, we will determine the 4-bit number as follows:

- The bit 0 ($2^0 = 1$) indicates the match to the top right neighbor chip.
- The bit 1 ($2^1 = 2$) indicates the match to the right neighbor chip.
- The bit 2 ($2^2 = 4$) indicates the match to the bottom right neighbor chips.
- The bit 3 ($2^3 = 8$) indicates the match to the bottom neighbor chip.

The positions of the matches can finally be written as the $R \times C$ matrix of the hexadecimal digits we get by converting the corresponding 4-bit numbers.

Write a program that will calculate number of ways Mary can move one match to another position so that each pair of chips is still connected and that the matches still do not intersect with each other.

Input

First line of the input contains two integers R and S ($2 \leq R, S \leq 500$) — number of rows and columns in the grid.

Each of next R lines contains S hexadecimal digits (digits '0' - '9' denote numbers from 1 to 9, digits 'A' - 'F' denote numbers from 10 to 15).

Output

Print one integer — number of ways to move one match such as each pair of chips is still connected and matches still do not intersect each other.

Example

standard input	standard output

2 2
20
30
	5

3 2
88
88
20
	20

3 3 8E8 FA8 220	43