A. Stairs and Elevators, 2 seconds, 256 megabytes, standard input, standard output

In the year of $30XX$ participants of some world programming championship live in a single large hotel. The hotel has n floors. Each floor has m sections with a single corridor connecting all of them. The sections are enumerated from 1 to m along the corridor, and all sections with equal numbers on different floors are located exactly one above the other. Thus, the hotel can be represented as a rectangle of height n and width m. We can denote sections with pairs of integers (i, j), where i is the floor, and j is the section number on the floor.

The guests can walk along the corridor on each floor, use stairs and elevators. Each stairs or elevator occupies all sections $(1, x)$, $(2, x)$, $\ldots$, (n, x) for some x between 1 and m. All sections not occupied with stairs or elevators contain guest rooms. It takes one time unit to move between neighboring sections on the same floor or to move one floor up or down using stairs. It takes one time unit to move up to v floors in any direction using an elevator. You can assume you don't have to wait for an elevator, and the time needed to enter or exit an elevator is negligible.

You are to process q queries. Each query is a question "what is the minimum time needed to go from a room in section (x_1, y_1) to a room in section (x_2, y_2)?"

Input

The first line contains five integers n, m, c_l, c_e, v ($2 \leq n, m \leq 10^8$, $0 \leq c_l, c_e \leq 10^5$, $1 \leq c_l + c_e \leq m - 1$, $1 \leq v \leq n - 1$) — the number of floors and section on each floor, the number of stairs, the number of elevators and the maximum speed of an elevator, respectively.

The second line contains c_l integers $l_1, \ldots, l_{c_l}$ in increasing order ($1 \leq l_i \leq m$), denoting the positions of the stairs. If $c_l = 0$, the second line is empty.

The third line contains c_e integers $e_1, \ldots, e_{c_e}$ in increasing order, denoting the elevators positions in the same format. It is guaranteed that all integers l_i and e_i are distinct.

The fourth line contains a single integer q ($1 \leq q \leq 10^5$) — the number of queries.

The next q lines describe queries. Each of these lines contains four integers x_1, y_1, x_2, y_2 ($1 \leq x_1, x_2 \leq n$, $1 \leq y_1, y_2 \leq m$) — the coordinates of starting and finishing sections for the query. It is guaranteed that the starting and finishing sections are distinct. It is also guaranteed that these sections contain guest rooms, i. e. y_1 and y_2 are not among l_i and e_i.

Output

Print q integers, one per line — the answers for the queries.

input
```
5 6 1 1 3
2
5
3
1 1 5 6
1 3 5 4
3 3 5 3
```

output
```
7
5
4
```

In the first query the optimal way is to go to the elevator in the 5-th section in four time units, use it to go to the fifth floor in two time units and go to the destination in one more time unit.

In the second query it is still optimal to use the elevator, but in the third query it is better to use the stairs in the section 2.

B. Resource Distribution, 2 seconds, 256 megabytes, standard input, standard output

One department of some software company has n servers of different specifications. Servers are indexed with consecutive integers from 1 to n. Suppose that the specifications of the j-th server may be expressed with a single integer number c_j of artificial resource units.

In order for production to work, it is needed to deploy two services S_1 and S_2 to process incoming requests using the servers of the department. Processing of incoming requests of service S_i takes x_i resource units.

The described situation happens in an advanced company, that is why each service may be deployed using not only one server, but several servers simultaneously. If service S_i is deployed using k_i servers, then the load is divided equally between these servers and each server requires only x_i / k_i (that may be a fractional number) resource units.

Each server may be left unused at all, or be used for deploying exactly one of the services (but not for two of them simultaneously). The service should not use more resources than the server provides.

Determine if it is possible to deploy both services using the given servers, and if yes, determine which servers should be used for deploying each of the services.

Input

The first line contains three integers n, x_1, x_2 ($2 \leq n \leq 300\,000$, $1 \leq x_1, x_2 \leq 10^9$) — the number of servers that the department may use, and resource units requirements for each of the services.

The second line contains n space-separated integers $c_1, c_2, \ldots, c_n$ ($1 \leq c_i \leq 10^9$) — the number of resource units provided by each of the servers.

Output

If it is impossible to deploy both services using the given servers, print the only word "No" (without the quotes).

Otherwise print the word "Yes" (without the quotes).

In the second line print two integers k_1 and k_2 ($1 \leq k_1, k_2 \leq n$) — the number of servers used for each of the services.

In the third line print k_1 integers, the indices of the servers that will be used for the first service.

In the fourth line print k_2 integers, the indices of the servers that will be used for the second service.

No index may appear twice among the indices you print in the last two lines. If there are several possible answers, it is allowed to print any of them.

input
```
6 8 16
3 5 2 9 8 7
```
output
```
Yes
3 2
1 2 6
5 4
```

input
```
4 20 32
21 11 11 12
```
output
```
Yes
1 3
1
2 3 4
```

input
```
4 11 32
5 5 16 16
```
output
```
No
```

input
```
5 12 20
7 8 4 11 9
```
output
```
No
```

In the first sample test each of the servers 1, 2 and 6 will will provide $8/3 = 2.(6)$ resource units and each of the servers 5, 4 will provide $16/2 = 8$ resource units.

In the second sample test the first server will provide 20 resource units and each of the remaining servers will provide $32/3 = 10.(6)$ resource units.

C. Big Secret,

2 seconds, 256 megabytes, standard input, standard output

Vitya has learned that the answer for The Ultimate Question of Life, the Universe, and Everything is not the integer ~~54~~ 42, but an increasing integer sequence $a_1, \dots, a_n$. In order to not reveal the secret earlier than needed, Vitya encrypted the answer and obtained the sequence $b_1, \dots, b_n$ using the following rules:

- $b_1 = a_1$;
- $b_i = a_i \oplus a_{i-1}$ for all i from 2 to n, where $x \oplus y$ is the bitwise XOR of x and y.

It is easy to see that the original sequence can be obtained using the rule $a_i = b_1 \oplus \dots \oplus b_i$.

However, some time later Vitya discovered that the integers b_i in the cypher got shuffled, and it can happen that when decrypted using the rule mentioned above, it can produce a sequence that is not increasing. In order to save his reputation in the scientific community, Vasya decided to find some permutation of integers b_i so that the sequence $a_i = b_1 \oplus \dots \oplus b_i$ is strictly increasing. Help him find such a permutation or determine that it is impossible.

Input

The first line contains a single integer n ($1 \leq n \leq 10^5$).

The second line contains n integers $b_1, \dots, b_n$ ($1 \leq b_i < 2^{60}$).

Output

If there are no valid permutations, print a single line containing "No".

Otherwise in the first line print the word "Yes", and in the second line print integers $b'_1, \dots, b'_n$ — a valid permutation of integers b_i. The unordered multisets $\{b_1, \dots, b_n\}$ and $\{b'_1, \dots, b'_n\}$ should be equal, i. e. for each integer x the number of occurrences of x in the first multiset should be equal to the number of occurrences of x in the second multiset. Apart from this, the sequence $a_i = b'_1 \oplus \dots \oplus b'_i$ should be strictly increasing.

If there are multiple answers, print any of them.

input
```
3
1 2 3
```
output
```
No
```

input
```
6
4 7 7 12 31 61
```
output
```
Yes
4 12 7 31 7 61
```

In the first example no permutation is valid.

In the second example the given answer lead to the sequence $a_1 = 4$, $a_2 = 8$, $a_3 = 15$, $a_4 = 16$, $a_5 = 23$, $a_6 = 42$.

D. Aztec Catacombs,

2 seconds, 256 megabytes, standard input, standard output

Indiana Jones found ancient Aztec catacombs containing a golden idol. The catacombs consists of n caves. Each pair of caves is connected with a two-way corridor that can be opened or closed. The entrance to the catacombs is in the cave 1, the idol and the exit are in the cave n.

When Indiana goes from a cave x to a cave y using an open corridor, all corridors connected to the cave x change their state: all open corridors become closed, all closed corridors become open. Indiana wants to go from cave 1 to cave n going through as small number of corridors as possible. Help him find the optimal path, or determine that it is impossible to get out of catacombs.

Input

The first line contains two integers n and m ($2 \leq n \leq 3 \cdot 10^5$, $0 \leq m \leq 3 \cdot 10^5$) — the number of caves and the number of open corridors at the initial moment.

The next m lines describe the open corridors. The i-th of these lines contains two integers u_i and v_i ($1 \le u_i, v_i \le n, u_i \ne v_i$) — the caves connected by the i-th open corridor. It is guaranteed that each unordered pair of caves is presented at most once.

Output

If there is a path to exit, in the first line print a single integer k — the minimum number of corridors Indians should pass through ($1 \le k \le 10^6$). In the second line print $k + 1$ integers $x_0, \dots, x_k$ — the number of caves in the order Indiana should visit them. The sequence $x_0, \dots, x_k$ should satisfy the following:

- $x_0 = 1, x_k = n$;
- for each i from 1 to k the corridor from x_{i-1} to x_i should be open at the moment Indiana walks along this corridor.

If there is no path, print a single integer -1.

We can show that if there is a path, there is a path consisting of no more than 10^6 corridors.

input
```
4 4
1 2
2 3
1 3
3 4
```
output
```
2
1 3 4
```

input
```
4 2
1 2
2 3
```
output
```
4
1 2 3 1 4
```

E. May Holidays, 5 seconds, 256 megabytes, standard input, standard output

It's May in Flatland, and there are m days in this month. Despite the fact that May Holidays are canceled long time ago, employees of some software company still have a habit of taking short or long vacations in May.

Of course, not all managers of the company like this. There are n employees in the company that form a tree-like structure of subordination: each employee has a unique integer id i between 1 and n, and each employee with id i (except the head manager whose id is 1) has exactly one direct manager with id p_i. The structure of subordination is not cyclic, i.e. if we start moving from any employee to his direct manager, then we will eventually reach the head manager. We define that an employee u is a subordinate of an employee v, if v is a direct manager of u, or the direct manager of u is a subordinate of v. Let s_i be the number of subordinates the i-th employee has (for example, $s_1 = n - 1$, because all employees except himself are subordinates of the head manager).

Each employee i has a bearing limit of t_i, which is an integer between 0 and s_i. It denotes the maximum number of the subordinates of the i-th employee being on vacation at the same moment that he can bear. If at some moment strictly more than t_i subordinates of the i-th employee are on vacation, and the i-th employee himself is not on a vacation, he becomes *displeased*.

In each of the m days of May exactly one event of the following two types happens: either one employee leaves on a vacation at the beginning of the day, or one employee returns from a vacation in the beginning of the day. You know the sequence of events in the following m days. Your task is to compute for each of the m days the number of displeased employees on that day.

Input

The first line contains two integers n and m ($2 \le n, m \le 10^5$) — the number of employees in the company and the number of days in May.

The second line contains $n - 1$ integers $p_2, p_3, \dots, p_n$ ($1 \le p_i \le n$), denoting the direct managers of employees.

The third line contains n integers $t_1, t_2, \dots, t_n$ ($0 \le t_i \le s_i$), denoting the bearing limits of empoyees.

The fourth line contains m integers $q_1, q_2, \dots, q_m$ ($1 \le |q_i| \le n$, $q_i \ne 0$), denoting the events. If q_i is positive, then the employee with id q_i leaves for a vacation starting from this day, if q_i is negative, then the employee $-q_i$ returns from a vacation starting from this day. In the beginning of May no employee is on vacation. It is guaranteed that if some employee leaves for a vacation, he is not on a vacation at the moment and vice versa.

Output

Print a sequence of m integers $a_1, a_2, \dots, a_m$, where a_i is the number of displeased employees on the i-th day.

input
```
7 8
4 5 1 1 5 5
0 0 0 1 2 0 0
2 6 3 7 -2 4 -3 1
```
output
```
1 1 1 2 2 2 1 0
```

input
```
5 6
1 2 3 4
4 0 0 1 0
1 5 2 3 -5 -1
```
output
```
0 2 1 0 0 0
```

In the first sample test after employee with id 2 leaves for a vacation at the first day, the head manager with id 1 becomes displeased as he does not want any of his subordinates to go for a vacation. At the fourth day employee with id 5 becomes displeased as his last remaining employee with id 7 leaves for a vacation. At the fifth day employee with id 2 returns from the vacation, but it does not affect the number of displeased employees as the employees 5 and 1 are still displeased. At the sixth day employee with id 3 returns back from the vacation, preventing the employee with id 5 from being displeased and at the last day the head manager with id 1 leaves for a vacation, leaving the company without the displeased people at all.

F. Parametric Circulation, 2 seconds, 256 megabytes, standard input, standard output

Vova has recently learned what a *circulaton* in a graph is. Recall the definition: let $G = (V, E)$ be a directed graph. A circulation f is such a collection of non-negative real numbers f_e ($e \in E$), that for each vertex $v \in V$ the following *conservation* condition holds:

$$\sum_{e \in \delta^-(v)} f_e = \sum_{e \in \delta^+(v)} f_e$$

where $\delta^+(v)$ is the set of edges that end in the vertex v, and $\delta^-(v)$ is the set of edges that start in the vertex v. In other words, for each vertex the total incoming flow should be equal to the total outcoming flow.

Let a lr-circulation be such a circulation f that for each edge the condition $l_e \leq f_e \leq r_e$ holds, where l_e and r_e for each edge $e \in E$ are two non-negative real numbers denoting the lower and upper bounds on the value of the circulation on this edge e.

Vova can't stop thinking about applications of a new topic. Right now he thinks about the following *natural* question: let the graph be fixed, and each value l_e and r_e be a linear function of a real variable t:

$$l_e(t) = a_e t + b_e$$

$$r_e(t) = c_e t + d_e$$

Note that t is the **same** for all edges.

Let t be chosen at random from uniform distribution on a segment $[0, 1]$. What is the probability of existence of lr-circulation in the graph?

Input

The first line contains two integers n, m ($1 \leq n \leq 1000$, $1 \leq m \leq 2000$).

Each of the next m lines describes edges of the graph in the format u_e, v_e, a_e, b_e, c_e, d_e ($1 \leq u_e, v_e \leq n$, $-10^4 \leq a_e, c_e \leq 10^4$, $0 \leq b_e, d_e \leq 10^4$), where u_e and v_e are the startpoint and the endpoint of the edge e, and the remaining 4 integers describe the linear functions for the upper and lower bound of circulation.

It is guaranteed that for any $t \in [0, 1]$ and for any edge $e \in E$ the following condition holds $0 \leq l_e(t) \leq r_e(t) \leq 10^4$.

Output

Print a single real integer — the probability of existence of lr-circulation in the graph, given that t is chosen uniformly at random from the segment $[0, 1]$. Your answer is considered correct if its absolute difference from jury's answer is not greater than 10^{-6}.

input

```
3 3
1 2 0 3 -4 7
2 3 -2 5 1 6
3 1 0 4 0 4
```

output

```
0.25
```

In the first example the conservation condition allows only circulations with equal values f_e for all three edges. The value of circulation on the last edge should be 4 whatever t is chosen, so the probability is

$$P(4 \in [3, -4t + 7] \ \& \ 4 \in [-2t + 5, t + 6]) = 0.25$$