A

Ant Colony

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Mole is hungry again. He found one ant colony, consisting of n ants, ordered in a row. Each ant i ($1 \le i \le n$) has a strength s_i.

In order to make his dinner more interesting, Mole organizes a version of «Hunger Games» for the ants. He chooses two numbers l and r ($1 \le l \le r \le n$) and each pair of ants with indices between l and r (inclusively) will fight. When two ants i and j fight, ant i gets one battle point only if s_i divides s_j (also, ant j gets one battle point only if s_j divides s_i).

After all fights have been finished, Mole makes the ranking. An ant i, with v_i battle points obtained, is going to be freed only if $v_i = r - l$, or in other words only if it took a point in every fight it participated. After that, Mole eats the rest of the ants. Note that there can be many ants freed or even none.

In order to choose the best sequence, Mole gives you t segments $[l_i, r_i]$ and asks for each of them how many ants is he going to eat if those ants fight.

Input

The first line contains one integer n ($1 \le n \le 10^5$), the size of the ant colony.

The second line contains n integers $s_1, s_2, \ldots, s_n$ ($1 \le s_i \le 10^9$), the strengths of the ants.

The third line contains one integer t ($1 \le t \le 10^5$), the number of test cases.

Each of the next t lines contains two integers l_i and r_i ($1 \le l_i \le r_i \le n$), describing one query.

Output

Print to the standard output t lines. The i-th line contains number of ants that Mole eats from the segment $[l_i, r_i]$.

Examples

input

```
5
1 3 2 4 2
4
1 5
2 5
3 5
4 5
```

output

```
4
4
1
1
```

Note

In the first test battle points for each ant are $v = [4, 0, 2, 0, 2]$, so ant number 1 is freed. Mole eats the ants $2, 3, 4, 5$.

In the second test case battle points are $v = [0, 2, 0, 2]$, so no ant is freed and all of them are eaten by Mole.

In the third test case battle points are $v = [2, 0, 2]$, so ants number 3 and 5 are freed. Mole eats

only the ant 4.

In the fourth test case battle points are $v = [0, 1]$, so ant number 5 is freed. Mole eats the ant 4.

B New Year and Entity Enumera

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given an integer m.

Let $M = 2^m - 1$.

You are also given a set of n integers denoted as the set T. The integers will be provided in base 2 as n binary strings of length m.

A set of integers S is called "good" if the following hold.

1. If $a \in S$, then $a \text{ XOR } M \in S$.
2. If $a, b \in S$, then $a \text{ AND } b \in S$
3. $T \subseteq S$
4. All elements of S are less than or equal to M.

Here, XOR and AND refer to the bitwise XOR and bitwise AND operators, respectively.

Count the number of good sets S, modulo $10^9 + 7$.

Input

The first line will contain two integers m and n ($1 \le m \le 1\,000$, $1 \le n \le min(2^m, 50)$).

The next n lines will contain the elements of T. Each line will contain exactly m zeros and ones. Elements of T will be distinct.

Output

Print a single integer, the number of good sets modulo $10^9 + 7$.

Examples

input

```
5 3
11010
00101
11000
```

output

```
4
```

input

```
30 2
010101010101010010101010101010
110110110110110011011011011011
```

output

```
860616440
```

Note

An example of a valid set S is {00000, 00101, 00010, 00111, 11000, 11010, 11101, 11111}.

C

Appleman and Tree

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Appleman has a tree with n vertices. Some of the vertices (at least one) are colored black and other vertices are colored white.

Consider a set consisting of k $(0 \le k < n)$ edges of Appleman's tree. If Appleman deletes these edges from the tree, then it will split into $(k+1)$ parts. Note, that each part will be a tree with colored vertices.

Now Appleman wonders, what is the number of sets splitting the tree in such a way that each resulting part will have exactly one black vertex? Find this number modulo 1000000007 $(10^9 + 7)$.

Input

The first line contains an integer n ($2 \le n \le 10^5$) — the number of tree vertices.

The second line contains the description of the tree: $n - 1$ integers $p_0, p_1, \ldots, p_{n-2}$ ($0 \le p_i \le i$). Where p_i means that there is an edge connecting vertex $(i+1)$ of the tree and vertex p_i. Consider tree vertices are numbered from 0 to $n - 1$.

The third line contains the description of the colors of the vertices: n integers $x_0, x_1, \ldots, x_{n-1}$ (x_i is either 0 or 1). If x_i is equal to 1, vertex i is colored black. Otherwise, vertex i is colored white.

Output

Output a single integer — the number of ways to split the tree modulo 1000000007 $(10^9 + 7)$.

Examples

input

```
3
0 0
0 1 1
```

output

```
2
```

input

```
6
0 1 1 0 4
1 1 0 0 1 0
```

output

```
1
```

input

```
10
0 1 2 1 4 4 4 0 8
0 0 0 1 0 1 1 0 0 1
```

output

```
27
```

D

Strange Regulations

`regulate.c`, `regulate.C`, `regulate.java`

Thank to cryptography, we are able to encrypt messages such that noone (except the intended recipient) is able to read them. However, encrypted messages are of no use if they do not actually *reach* the recipient. These days, computer network is the most typical mean to send such messages. In this problem, we will study the issues the networking providers have to solve. And remember: since the message is encrypted, we do not need to care about the network privacy anymore.

The network cables joining computers (servers) belong to different companies. A new anti-monopoly legislation prevents any company from owning more than two cables from each server. Furthermore, to avoid wasting resources, there is also a law specifying that the cable system owned by any single company cannot be redundant, i.e., removal of *any* of the cables will disconnect some two previously connected servers. Since the companies buy and sell the cables all the time, it is quite difficult to enforce these regulations. Your task is to write a program that does so.

Input Specification

The input contains several instances. The first line of each instance contains four integers N, M, C and T separated by spaces — the number of servers ($1 \leq N \leq 8\,000$), the number of cables ($0 \leq M \leq 100\,000$), the number of companies ($1 \leq C \leq 100$), and the number of cable-selling transactions ($0 \leq T \leq 100\,000$), respectively.

The following M lines describe the cables. Each of them contains three integers S_{j1}, S_{j2} and K_j, separated by spaces, giving the numbers of the servers S_{j1} and S_{j2} ($1 \leq S_{j1} < S_{j2} \leq n$) joined by that cable and the number of the company K_j ($1 \leq K_j \leq C$) initially owning the cable. For each pair of servers, there is at most one cable joining them. The initial state satisfies the regulations, i.e., each company owns at most two cables incident with each server, and the system of cables owned by a single company has no cycles.

Finally, each of the next T lines contains integers S_{i1}, S_{i2} and K_i describing one transaction in which the company K_i ($1 \leq K_i \leq C$) tries to buy a cable between servers S_{i1} and S_{i2} ($1 \leq S_{i1} < S_{i2} \leq N$).

The last instance is followed by a line containing four zeros.

Output Specification

For each input instance, output T lines describing the outcome of the transactions. The possible outcomes are

- "`No such cable.`" if the pair of servers is not joined by a cable,
- "`Already owned.`" if the cable is already owned by the company K_i,
- "`Forbidden: monopoly.`" if the company K_i already owns two cables at S_{i1} or S_{i2},
- "`Forbidden: redundant.`" if K_i owns at most one cable at each of S_{i1} and S_{i2}, but granting the ownership would create a cycle of cables owned by K_i,
- "`Sold.`" if none of the above restrictions apply. In this case, the ownership of the cable between S_{i1} and S_{i2} changes to K_i for the purpose of further transactions.

  Print one empty line after each instance.

Sample Input

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

Output for Sample Input

```
Sold.
Already owned.
Forbidden: monopoly.
Forbidden: redundant.
No such cable.

Already owned.
```

E

Restaurant

time limit per test: 4 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

A restaurant received n orders for the rental. Each rental order reserve the restaurant for a continuous period of time, the i-th order is characterized by two time values — the start time l_i and the finish time r_i ($l_i \le r_i$).

Restaurant management can accept and reject orders. What is the maximal number of orders the restaurant can accept?

No two accepted orders can intersect, i.e. they can't share even a moment of time. If one order ends in the moment other starts, they can't be accepted both.

Input

The first line contains integer number n ($1 \le n \le 5 \cdot 10^5$) — number of orders. The following n lines contain integer values l_i and r_i each ($1 \le l_i \le r_i \le 10^9$).

Output

Print the maximal number of orders that can be accepted.

Examples

input

```
2
7 11
4 7
```

output

```
1
```

input

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

output

```
3
```

input

```
6
4 8
1 5
4 7
2 5
1 3
6 8
```

output

```
2
```

F

Mr. Bender and Square

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Mr. Bender has a digital table of size $n \times n$, each cell can be switched on or off. He wants the field to have at least c switched on squares. When this condition is fulfilled, Mr Bender will be happy.

We'll consider the table rows numbered from top to bottom from 1 to n, and the columns — numbered from left to right from 1 to n. Initially there is exactly one switched on cell with coordinates (x, y) (x is the row number, y is the column number), and all other cells are switched off. Then each second we switch on the cells that are off but have the side-adjacent cells that are on.

For a cell with coordinates (x, y) the side-adjacent cells are cells with coordinates $(x - 1, y)$, $(x + 1, y)$, $(x, y - 1)$, $(x, y + 1)$.

In how many seconds will Mr. Bender get happy?

Input

The first line contains four space-separated integers n, x, y, c $(1 \le n, c \le 10^9; 1 \le x, y \le n; c \le n^2)$.

Output

In a single line print a single integer — the answer to the problem.

Examples

input

```
6 4 3 1
```

output

```
0
```

input

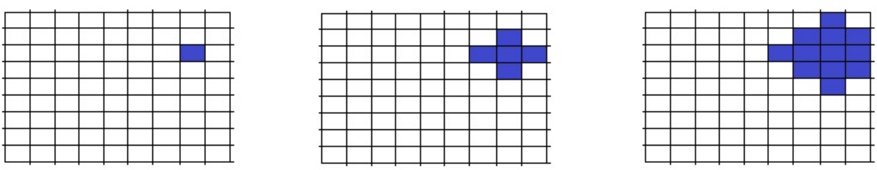
```
9 3 8 10
```

output

```
2
```

Note

Initially the first test has one painted cell, so the answer is 0. In the second test all events will go as is shown on the figure.



.

G Maximum Distance

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Chouti was tired of the tedious homework, so he opened up an old programming problem he created years ago.

You are given a connected undirected graph with n vertices and m weighted edges. There are k special vertices: $x_1, x_2, \ldots, x_k$.

Let's define the cost of the path as the **maximum** weight of the edges in it. And the *distance* between two vertexes as the **minimum** cost of the paths connecting them.

For each special vertex, find another special vertex which is farthest from it (in terms of the previous paragraph, i.e. the corresponding *distance* is maximum possible) and output the distance between them.

The original constraints are really small so he thought the problem was boring. Now, he raises the constraints and hopes you can solve it for him.

Input

The first line contains three integers n, m and k ($2 \le k \le n \le 10^5$, $n - 1 \le m \le 10^5$) — the number of vertices, the number of edges and the number of special vertices.

The second line contains k distinct integers $x_1, x_2, \ldots, x_k$ ($1 \le x_i \le n$).

Each of the following m lines contains three integers u, v and w ($1 \le u, v \le n, 1 \le w \le 10^9$), denoting there is an edge between u and v of weight w. The given graph is undirected, so an edge (u, v) can be used in the both directions.

The graph may have multiple edges and self-loops.

It is guaranteed, that the graph is connected.

Output

The first and only line should contain k integers. The i-th integer is the distance between x_i and the farthest special vertex from it.

Examples

input

```
2 3 2
2 1
1 2 3
1 2 2
2 2 1
```

output

```
2 2
```

input

```
4 5 3
1 2 3
1 2 5
4 2 1
2 3 2
1 4 4
1 3 3
```

output

```
3 3 3
```

Note

In the first example, the distance between vertex 1 and 2 equals to 2 because one can walk through the edge of weight 2 connecting them. So the distance to the farthest node for both 1 and 2 equals to 2.

In the second example, one can find that distance between 1 and 2, distance between 1 and 3 are both 3 and the distance between 2 and 3 is 2.

The graph may have multiple edges between and self-loops, as in the first example.

H

Meetings

time limit per test: 1 second

memory limit per test: 64 megabytes

input: input.txt

output: output.txt

Two cities A and B are connected by a straight road that is exactly l meters long. At the initial moment of time a cyclist starts moving from city A to city B at a speed v_1 meters/second, and a pedestrian starts moving from city B to city A at a speed v_2 meters/second. When one of them reaches a city, the road ends, so the person has to turn around and start moving in the opposite direction by the same road, keeping the original speed. As a result, the cyclist and the pedestrian are traveling between cities A and B indefinitely.

Your task is to calculate the number of times they will meet during the first t seconds. If they meet in exactly t seconds after the initial moment of time, this meeting should also be counted.

Input

The only line of input contains four integer numbers: l, v_1, v_2 and t. All numbers are between 1 and 10^9, inclusively.

Output

Print a single integer — the number of times the cyclist and the pedestrian will meet during the first t seconds.

Examples

input

```
1000 10 1 200
```

output

```
2
```

input

```
4 4 3 4
```

output

```
4
```

I

Dima and Horses

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Dima came to the horse land. There are n horses living in the land. Each horse in the horse land has several enemies (enmity is a symmetric relationship). The horse land isn't very hostile, so the number of enemies of each horse is at most 3.

Right now the horse land is going through an election campaign. So the horses trusted Dima to split them into two parts. At that the horses want the following condition to hold: a horse shouldn't have more than one enemy in its party.

Help Dima split the horses into parties. Note that one of the parties can turn out to be empty.

Input

The first line contains two integers n, m $(1 \le n \le 3 \cdot 10^5; 0 \le m \le \min(3 \cdot 10^5, \frac{n(n-1)}{2}))$ — the number of horses in the horse land and the number of enemy pairs.

Next m lines define the enemy pairs. The i-th line contains integers a_i, b_i $(1 \le a_i, b_i \le n;\ a_i \ne b_i)$, which mean that horse a_i is the enemy of horse b_i.

Consider the horses indexed in some way from 1 to n. It is guaranteed that each horse has at most three enemies. No pair of enemies occurs more than once in the input.

Output

Print a line, consisting of n characters: the i-th character of the line must equal "`0`", if the horse number i needs to go to the first party, otherwise this character should equal "`1`".

If there isn't a way to divide the horses as required, print `-1`.

Examples

input

```
3 3
1 2
3 2
3 1
```

output

```
100
```

input

```
2 1
2 1
```

output

```
00
```

input

```
10 6
1 2
1 3
1 4
2 3
2 4
3 4
```

output
0110000000

J

Vasya and a Tree

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Vasya has a tree consisting of n vertices with root in vertex 1. At first all vertices has 0 written on it.

Let $d(i, j)$ be the distance between vertices i and j, i.e. number of edges in the shortest path from i to j. Also, let's denote k-subtree of vertex x — set of vertices y such that next two conditions are met:

- x is the ancestor of y (each vertex is the ancestor of itself);
- $d(x, y) \le k$.

Vasya needs you to process m queries. The i-th query is a triple v_i, d_i and x_i. For each query Vasya adds value x_i to each vertex from d_i-subtree of v_i.

Report to Vasya all values, written on vertices of the tree after processing all queries.

Input

The first line contains single integer n ($1 \le n \le 3 \cdot 10^5$) — number of vertices in the tree.

Each of next $n - 1$ lines contains two integers x and y ($1 \le x, y \le n$) — edge between vertices x and y. It is guarantied that given graph is a tree.

Next line contains single integer m ($1 \le m \le 3 \cdot 10^5$) — number of queries.

Each of next m lines contains three integers v_i, d_i, x_i ($1 \le v_i \le n, 0 \le d_i \le 10^9$, $1 \le x_i \le 10^9$) — description of the i-th query.

Output

Print n integers. The i-th integers is the value, written in the i-th vertex after processing all queries.

Examples

input

```
5
1 2
1 3
2 4
2 5
3
1 1 1
2 0 10
4 10 100
```

output

```
1 11 1 100 0
```

input

```
5
2 3
2 1
5 4
3 4
5
2 0 4
3 10 1
1 2 3
```

```
2 3 10
1 1 7
```

output

```
10 24 14 11 11
```

Note

In the first exapmle initial values in vertices are $0, 0, 0, 0, 0$. After the first query values will be equal to $1, 1, 1, 0, 0$. After the second query values will be equal to $1, 11, 1, 0, 0$. After the third query values will be equal to $1, 11, 1, 100, 0$.

K Case of Fugitive

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Andrewid the Android is a galaxy-famous detective. He is now chasing a criminal hiding on the planet Oxa-5, the planet almost fully covered with water.

The only dry land there is an archipelago of n narrow islands located in a row. For more comfort let's represent them as non-intersecting segments on a straight line: island i has coordinates $[l_i, r_i]$, besides, $r_i < l_{i+1}$ for $1 \le i \le n - 1$.

To reach the goal, Andrewid needs to place a bridge between each pair of **adjacent** islands. A bridge of length a can be placed between the i-th and the $(i+1)$-th islads, if there are such coordinates of x and y, that $l_i \le x \le r_i$, $l_{i+1} \le y \le r_{i+1}$ and $y - x = a$.

The detective was supplied with m bridges, each bridge can be used at most once. Help him determine whether the bridges he got are enough to connect each pair of adjacent islands.

Input

The first line contains integers n $(2 \le n \le 2 \cdot 10^5)$ and m $(1 \le m \le 2 \cdot 10^5)$ — the number of islands and bridges.

Next n lines each contain two integers l_i and r_i $(1 \le l_i \le r_i \le 10^{18})$ — the coordinates of the island endpoints.

The last line contains m **integer** numbers $a_1, a_2, ..., a_m$ $(1 \le a_i \le 10^{18})$ — the lengths of the bridges that Andrewid got.

Output

If it is impossible to place a bridge between each pair of adjacent islands in the required manner, print on a single line "`No`" (without the quotes), otherwise print in the first line "`Yes`" (without the quotes), and in the second line print $n - 1$ numbers $b_1, b_2, ..., b_{n-1}$, which mean that between islands i and $i+1$ there must be used a bridge number b_i.

If there are multiple correct answers, print any of them. Note that in this problem it is necessary to print "`Yes`" and "`No`" in correct case.

Examples

input

```
4 4
1 4
7 8
9 10
12 14
4 5 3 8
```

output

```
Yes
2 3 1
```

input

```
2 2
11 14
17 18
2 9
```

output

```
No
```

input

```
2 1
1 1
1000000000000000000 1000000000000000000
999999999999999999
```

output

```
Yes
1
```

Note

In the first sample test you can, for example, place the second bridge between points 3 and 8, place the third bridge between points 7 and 10 and place the first bridge between points 10 and 14.

In the second sample test the first bridge is too short and the second bridge is too long, so the solution doesn't exist.

L. Number Busters

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Arthur and Alexander are number busters. Today they've got a competition.

Arthur took a group of four integers a, b, w, x $(0 \le b < w, 0 < x < w)$ and Alexander took integer c. Arthur and Alexander use distinct approaches to number bustings. Alexander is just a regular guy. Each second, he subtracts one from his number. In other words, he performs the assignment: $c = c - 1$. Arthur is a sophisticated guy. Each second Arthur performs a complex operation, described as follows: if $b \ge x$, perform the assignment $b = b - x$, if $b < x$, then perform two consecutive assignments $a = a - 1;\ b = w - (x - b)$.

You've got numbers a, b, w, x, c. Determine when Alexander gets ahead of Arthur if both guys start performing the operations at the same time. Assume that Alexander got ahead of Arthur if $c \le a$.

Input

The first line contains integers a, b, w, x, c $(1 \le a \le 2 \cdot 10^9, 1 \le w \le 1000, 0 \le b < w, 0 < x < w, 1 \le c \le 2 \cdot 10^9)$.

Output

Print a single integer — the minimum time in seconds Alexander needs to get ahead of Arthur. You can prove that the described situation always occurs within the problem's limits.

Examples

input

```
4 2 3 1 6
```

output

```
2
```

input

```
4 2 3 1 7
```

output

```
4
```

input

```
1 2 3 2 6
```

output

```
13
```

input

```
1 1 2 1 1
```

output

```
0
```

Frightful Formula

M

Time limit: 10 s
Memory limit: 512 MiB

A *frightful matrix* is a square matrix of order n where the first row and the first column are explicitly specified, while the other elements are calculated using a *frightful formula* which is, actually, a simple recursive rule.

Given two integer sequences l and t, both of size n, as well as integer parameters a, b and c, the frightful matrix F is defined as follows:

- The first column of the matrix is the sequence l:

$$F[k,\ 1] = l_k.$$

- The first row of the matrix is the sequence t:

$$F[1,\ k] = t_k.$$

- Other elements are calculated using a recursive formula:

$$F[i,\ j] = a * F[i,\ j-1] + b * F[i-1,\ j] + c.$$

Given a frightful matrix, find the value of the element $F[n, n]$ modulo $10^6 + 3$.

Input

The first line contains four integers n, a, b and c ($2 \le n \le 200\,000$, $0 \le a,\ b,\ c \le 10^6$) – the size of the matrix and the recursion parameters, as described in the problem statement.

The two following lines contain integers $l_1, \ldots, l_n$ and $t_1, \ldots, t_n$, respectively ($l_1 = t_1$, $0 \le l_k,\ t_k \le 10^6$).

Output

Output a single integer – the value of $F[n, n]$ modulo $10^6 + 3$.

Example

input

```
3 0 0 0
0 0 2
0 3 0
```

output

```
0
```

input

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

output

```
41817
```