A. Duff and Meat

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

Duff is addicted to meat! Malek wants to keep her happy for n days. In order to be happy in i-th day, she needs to eat exactly a_i kilograms of meat.

There is a big shop uptown and Malek wants to buy meat for her from there. In i-th day, they sell meat for p_i dollars per kilogram. Malek knows all numbers $a_1, ..., a_n$ and $p_1, ..., p_n$. In each day, he can buy arbitrary amount of meat, also he can keep some meat he has for the future.

Malek is a little tired from cooking meat, so he asked for your help. Help him to minimize the total money he spends to keep Duff happy for n days.

Input

The first line of input contains integer n ($1 \leq n \leq 10^5$), the number of days.

In the next n lines, i-th line contains two integers a_i and p_i ($1 \leq a_i, p_i \leq 100$), the amount of meat Duff needs and the cost of meat in that day.

Output

Print the minimum money needed to keep Duff happy for n days, in one line.

Examples

input

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

output

```
10
```

input

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

output

```
8
```

Note

In the first sample case: An optimal way would be to buy 1 kg on the first day, 2 kg on the second day and 3 kg on the third day.

In the second sample case: An optimal way would be to buy 1 kg on the first day and 5 kg (needed meat for the second and third day) on the second day.

B. Duff in Love

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

Duff is in love with lovely numbers! A positive integer x is called *lovely* if and only if there is no such positive integer $a > 1$ such that a^2 is a divisor of x.

Malek has a number store! In his store, he has only divisors of positive integer n (and he has all of them). As a birthday present, Malek wants to give her a *lovely* number from his store. He wants this number to be as big as possible.

Malek always had issues in math, so he asked for your help. Please tell him what is the biggest lovely number in his store.

Input

The first and only line of input contains one integer, n ($1 \leq n \leq 10^{12}$).

Output

Print the answer in one line.

Examples

input

```
10
```

output

```
10
```

input

```
12
```

output

```
6
```

Note

In first sample case, there are numbers 1, 2, 5 and 10 in the shop. 10 isn't divisible by any perfect square, so 10 is *lovely*.

In second sample case, there are numbers 1, 2, 3, 4, 6 and 12 in the shop. 12 is divisible by $4 = 2^2$, so 12 is not *lovely*, while 6 is indeed *lovely*.

C. Duff and Weight Lifting

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

Recently, Duff has been practicing weight lifting. As a hard practice, Malek gave her a task. He gave her a sequence of weights. Weight of i-th of them is 2^{w_i} pounds. In each step, Duff can lift some of the remaining weights and throw them away. She does this until there's no more weight left. Malek asked her to minimize the number of steps.

Duff is a competitive programming fan. That's why in each step, she can only lift and throw away a sequence of weights $2^{a_1}, \ldots, 2^{a_k}$ if and only if there exists a non-negative integer x such that $2^{a_1} + 2^{a_2} + \ldots + 2^{a_k} = 2^x$, i. e. the sum of those numbers is a power of two.

Duff is a competitive programming fan, but not a programmer. That's why she asked for your help. Help her minimize the number of steps.

Input

The first line of input contains integer n ($1 \le n \le 10^6$), the number of weights.

The second line contains n integers $w_1, \ldots, w_n$ separated by spaces ($0 \le w_i \le 10^6$ for each $1 \le i \le n$), the powers of two forming the weights values.

Output

Print the minimum number of steps in a single line.

Examples

input

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

output

```
2
```

input

```
4
0 1 2 3
```

output

```
4
```

Note

In the first sample case: One optimal way would be to throw away the first three in the first step and the rest in the second step. Also, it's not possible to do it in one step because their sum is not a power of two.

In the second sample case: The only optimal way is to throw away one weight in each step. It's not possible to do it in less than 4 steps because there's no subset of weights with more than one weight and sum equal to a power of two.

D. Duff in Beach

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

While Duff was resting in the beach, she accidentally found a strange array $b_0, b_1, \ldots, b_{l-1}$ consisting of l positive integers. This array was strange because it was extremely long, but there was another (maybe shorter) array, $a_0, \ldots, a_{n-1}$ that b can be build from a with formula: $b_i = a_{i \bmod n}$ where $a \bmod b$ denoted the remainder of dividing a by b.

Duff is so curious, she wants to know the number of subsequences of b like $b_{i_1}, b_{i_2}, \ldots, b_{i_x}$ ($0 \le i_1 < i_2 < \ldots < i_x < l$), such that:

- $1 \le x \le k$
- For each $1 \le j \le x - 1$, $\lfloor \frac{i_j}{n} \rfloor + 1 = \lfloor \frac{i_{j+1}}{n} \rfloor$
- For each $1 \le j \le x - 1$, $b_{i_j} \le b_{i_{j+1}}$. i.e this subsequence is non-decreasing.

Since this number can be very large, she want to know it modulo $10^9 + 7$.

Duff is not a programmer, and Malek is unavailable at the moment. So she asked for your help. Please tell her this number.

Input

The first line of input contains three integers, n, l and k ($1 \le n, k, n \times k \le 10^6$ and $1 \le l \le 10^{18}$).

The second line contains n space separated integers, $a_0, a_1, \ldots, a_{n-1}$ ($1 \le a_i \le 10^9$ for each $0 \le i \le n - 1$).

Output

Print the answer modulo $1\,000\,000\,007$ in one line.

Examples

input

```
3 5 3
5 9 1
```

output

```
10
```

input

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

output

```
25
```

Note

In the first sample case, $b = \langle 5, 9, 1, 5, 9 \rangle$. So all such sequences are: $\langle b_0 \rangle = \langle 5 \rangle$, $\langle b_1 \rangle = \langle 9 \rangle$, $\langle b_2 \rangle = \langle 1 \rangle$, $\langle b_3 \rangle = \langle 5 \rangle$, $\langle b_4 \rangle = \langle 9 \rangle$, $\langle b_0, b_3 \rangle = \langle 5, 5 \rangle$, $\langle b_0, b_4 \rangle = \langle 5, 9 \rangle$, $\langle b_1, b_4 \rangle = \langle 9, 9 \rangle$, $\langle b_2, b_3 \rangle = \langle 1, 5 \rangle$ and $\langle b_2, b_4 \rangle = \langle 1, 9 \rangle$.

E. Duff in the Army

time limit per test: 4 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Recently Duff has been a soldier in the army. Malek is her commander.

Their country, Andarz Gu has n cities (numbered from 1 to n) and $n - 1$ bidirectional roads. Each road connects two different cities. There exist a unique path between any two cities.

There are also m people living in Andarz Gu (numbered from 1 to m). Each person has and ID number. ID number of $i - th$ person is i and he/she lives in city number c_i. Note that there may be more than one person in a city, also there may be no people living in the city.

Malek loves to order. That's why he asks Duff to answer to q queries. In each query, he gives her numbers v, u and a.

To answer a query:

Assume there are x people living in the cities lying on the path from city v to city u. Assume these people's IDs are $p_1, p_2, ..., p_x$ in increasing order.

If $k = min(x, a)$, then Duff should tell Malek numbers $k, p_1, p_2, ..., p_k$ in this order. In the other words, Malek wants to know a minimums on that path (or less, if there are less than a people).

Duff is very busy at the moment, so she asked you to help her and answer the queries.

Input

The first line of input contains three integers, n, m and q ($1 \le n, m, q \le 10^5$).

The next $n - 1$ lines contain the roads. Each line contains two integers v and u, endpoints of a road ($1 \le v, u \le n$, $v \ne u$).

Next line contains m integers $c_1, c_2, ..., c_m$ separated by spaces ($1 \le c_i \le n$ for each $1 \le i \le m$).

Next q lines contain the queries. Each of them contains three integers, v, u and a ($1 \le v, u \le n$ and $1 \le a \le 10$).

Output

For each query, print numbers $k, p_1, p_2, ..., p_k$ separated by spaces in one line.

Examples

input

```
5 4 5
1 3
1 2
1 4
4 5
2 1 4 3
4 5 6
1 5 2
5 5 10
2 3 3
5 3 1
```

output

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

Note

Graph of Andarz Gu in the sample case is as follows (ID of people in each city are written next to them):

F. Duff in Mafia

time limit per test: 6 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Duff is one if the heads of Mafia in her country, Andarz Gu. Andarz Gu has n cities (numbered from 1 to n) connected by m bidirectional roads (numbered by 1 to m).

Each road has a destructing time, and a color. i-th road connects cities v_i and u_i and its color is c_i and its destructing time is t_i.

Mafia wants to destruct a *matching* in Andarz Gu. A *matching* is a subset of roads such that no two roads in this subset has common endpoint. They can destruct these roads in parallel, i. e. the total destruction time is a maximum over destruction times of all selected roads.

They want two conditions to be satisfied:

1. The remaining roads form a *proper coloring*.
2. Destructing time of this matching is minimized.

The remaining roads after destructing this matching form a *proper coloring* if and only if no two roads of the same color have same endpoint, or, in the other words, edges of each color should form a *matching*.

There is no programmer in Mafia. That's why Duff asked for your help. Please help her and determine which matching to destruct in order to satisfied those conditions (or state that this is not possible).

Input

The first line of input contains two integers n and m ($2 \le n \le 5 \times 10^4$ and $1 \le m \le 5 \times 10^4$), number of cities and number of roads in the country.

The next m lines contain the the roads. $i-th$ of them contains four integers v_i, u_i, c_i and t_i ($1 \le v_i, u_i \le n$, $v_i \ne u_i$ and $1 \le c_i, t_i \le 10^9$ for each $1 \le i \le m$).

Output

In the first line of input, print "Yes" (without quotes) if satisfying the first condition is possible and "No" (without quotes) otherwise.

If it is possible, then you have to print two integers t and k in the second line, the minimum destructing time and the number of roads in the matching ($0 \le k \le \lfloor \frac{n}{2} \rfloor$).

In the third line print k distinct integers separated by spaces, indices of the roads in the matching in any order. Roads are numbered starting from one in order of their appearance in the input.

If there's more than one solution, print any of them.

Examples

input

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

output

```
Yes
3 2
4 5
```

input

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

output

```
No
```

Note

Graph of Andarz Gu in the first sample case is as follows:

A solution would be to destruct the roads with crosses.

Graph of Andarz Gu in the second sample case is as follows: