A. The Meaningless Game

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

Slastyona and her loyal dog Pushok are playing a meaningless *game* that is indeed very interesting.

The *game* consists of multiple *rounds*. Its rules are very simple: in each round, a natural number k is chosen. Then, the one who says (or barks) it faster than the other wins the *round*. After that, the winner's score is multiplied by k^2, and the loser's score is multiplied by k. In the beginning of the *game*, both Slastyona and Pushok have scores equal to one.

Unfortunately, Slastyona had lost her notepad where the history of all n *games* was recorded. She managed to recall the final results for each games, though, but all of her memories of them are vague. Help Slastyona verify their correctness, or, to put it another way, for each given pair of scores determine whether it was possible for a game to finish with such result or not.

Input

In the first string, the number of games n $(1 \le n \le 350000)$ is given.

Each *game* is represented by a pair of scores a, b $(1 \le a, b \le 10^9)$ – the results of Slastyona and Pushok, correspondingly.

Output

For each pair of scores, answer "`Yes`" if it's possible for a game to finish with given score, and "`No`" otherwise.

You can output each letter in arbitrary case (upper or lower).

Example

input

```
6
2 4
75 45
8 8
16 16
247 994
1000000000 1000000
```

output

```
Yes
Yes
Yes
No
No
Yes
```

Note

First *game* might have been consisted of one round, in which the number 2 would have been chosen and Pushok would have won.

The second *game* needs exactly two rounds to finish with such result: in the first one, Slastyona would have said the number 5, and in the second one, Pushok would have barked the number 3.

B. The Bakery

time limit per test: 2.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Some time ago Slastyona the Sweetmaid decided to open her own bakery! She bought required ingredients and a wonder-oven which can bake several types of cakes, and opened the bakery.

Soon the expenses started to overcome the income, so Slastyona decided to study the sweets market. She learned it's profitable to pack cakes in boxes, and that the more **distinct** cake types a box contains (let's denote this number as the *value* of the box), the higher price it has.

She needs to change the production technology! The problem is that the oven chooses the cake types on its own and Slastyona can't affect it. However, she knows the types and order of n cakes the oven is going to bake today. Slastyona has to pack exactly k boxes with cakes today, and she has to put in each box several (at least one) cakes the oven produced one **right after another** (in other words, she has to put in a box a continuous segment of cakes).

Slastyona wants to maximize the total value of all boxes with cakes. Help her determine this maximum possible total value.

Input

The first line contains two integers n and k $(1 \le n \le 35000,\ 1 \le k \le min(n, 50))$ – the number of cakes and the number of boxes, respectively.

The second line contains n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le n)$ – the types of cakes in the order the oven bakes them.

Output

Print the only integer – the maximum total value of all boxes with cakes.

Examples

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

input
```
7 2
1 3 3 1 4 4 4
```
output
```
5
```

input
```
8 3
7 7 8 7 7 8 1 7
```
output
```
6
```

Note

In the first example Slastyona has only one box. She has to put all cakes in it, so that there are two types of cakes in the box, so the value is equal to 2.

In the second example it is profitable to put the first two cakes in the first box, and all the rest in the second. There are two distinct types in the first box, and three in the second box then, so the total value is 5.

C. Ever-Hungry Krakozyabra

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Recently, a wild Krakozyabra appeared at Jelly Castle. It is, truth to be said, always eager to have something for dinner.

Its favorite meal is natural numbers (typically served with honey sauce), or, to be more precise, the zeros in their corresponding decimal representations. As for other digits, Krakozyabra dislikes them; moreover, they often cause it indigestion! So, as a necessary precaution, Krakozyabra prefers to sort the digits of a number in non-descending order before proceeding to feast. Then, the leading zeros of the resulting number are eaten and the remaining part is discarded as an *inedible tail*.

For example, if Krakozyabra is to have the number 57040 for dinner, its *inedible tail* would be the number 457.

Slastyona is not really fond of the idea of Krakozyabra living in her castle. Hovewer, her natural hospitality prevents her from leaving her guest without food. Slastyona has a range of natural numbers from L to R, which she is going to feed the guest with. Help her determine how many distinct *inedible tails* are going to be discarded by Krakozyabra by the end of the dinner.

Input

In the first and only string, the numbers L and R are given – the boundaries of the range $(1 \le L \le R \le 10^{18})$.

Output

Output the sole number – the answer for the problem.

Examples

input

```
1 10
```

output

```
9
```

input

```
40 57
```

output

```
17
```

input

```
157 165
```

output

```
9
```

Note

In the first sample case, the *inedible tails* are the numbers from 1 to 9. Note that 10 and 1 have the same *inedible tail* – the number 1.

In the second sample case, each number has a unique *inedible tail*, except for the pair 45, 54. The answer to this sample case is going to be $(57 - 40 + 1) - 1 = 17$.

D. Red-Black Cobweb

time limit per test: 6 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Slastyona likes to watch life of nearby grove's dwellers. This time she watches a strange red-black spider sitting at the center of a huge cobweb.

The cobweb is a set of n nodes connected by threads, each of the treads is either red of black. Using these threads, the spider can move between nodes. No thread connects a node to itself, and between any two nodes there is a unique sequence of threads connecting them.

Slastyona decided to study some special qualities of the cobweb. She noticed that each of the threads has a value of *clamminess* x.

However, Slastyona is mostly interested in *jelliness* of the cobweb. Consider those of the shortest paths between each pair of nodes on which the numbers of red and black threads differ at most twice. For each such path compute the product of the clamminess of threads on the path.The jelliness of the cobweb is the product of all obtained values among all paths. Those paths that differ by direction only are counted only once.

Of course, this number can be huge, so Slastyona asks you to compute the jelliness of the given cobweb and print the answer modulo $10^9 + 7$.

Input

The first line contains the number of nodes n $(2 \le n \le 10^5)$.

The next $n - 1$ lines contain four integers each, denoting the i-th thread of the cobweb: the nodes it connects u_i, v_i $(1 \le u_i \le n, 1 \le v_i \le n)$, the *clamminess* of the thread x_i $(1 \le x \le 10^9 + 6)$, and the color of the thread c_i ($c_i \in \{0, 1\}$). The red color is denoted by 0, and the black color is denoted by 1.

Output

Print single integer the *jelliness* of the cobweb modulo $10^9 + 7$. If there are no paths such that the numbers of red and black threads differ at most twice, print 1.

Examples

input

```
5
1 2 9 0
2 3 5 1
2 4 5 0
2 5 5 1
```

output

```
1265625
```

input

```
8
1 2 7 1
2 3 4 1
3 4 19 1
5 1 2 0
6 2 3 0
7 3 3 0
8 4 4 0
```

output

```
452841614
```

Note

In the first example there are 4 pairs of nodes such that the numbers of threads of both colors on them differ at most twice. There pairs are $(1, 3)$ with product of clamminess equal to 45, $(1, 5)$ with product of clamminess equal to 45, $(3, 4)$ with product of clamminess equal to 25 and $(4, 5)$ with product of clamminess equal to 25. The *jelliness* of the cobweb is equal to 1265625.

E. Caramel Clouds

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

It is well-known that the best decoration for a flower bed in Sweetland are vanilla muffins. Seedlings of this plant need sun to grow up. Slastyona has m seedlings, and the j-th seedling needs at least k_j minutes of sunlight to grow up.

Most of the time it's sunny in Sweetland, but sometimes some caramel clouds come, the i-th of which will appear at time moment (minute) l_i and disappear at time moment r_i. Of course, the clouds make shadows, and the seedlings can't grow when there is at least one cloud veiling the sun.

Slastyona wants to grow up her muffins as fast as possible. She has exactly C candies, which is the main currency in Sweetland.

One can dispel any cloud by paying c_i candies. However, in order to comply with Sweetland's Department of Meteorology regulations, **one can't dispel more than two clouds**.

Slastyona hasn't decided yet which of the m seedlings will be planted at the princess' garden, so she needs your help. For each seedling determine the earliest moment it can grow up if Slastyona won't break the law and won't spend more candies than she has. Note that each of the seedlings is considered independently.

The seedlings start to grow at time moment 0.

Input

The first line contains two integers n and C $(0 \le n \le 3 \cdot 10^5,\ 0 \le C \le 10^9)$ – the number of caramel clouds and the number of candies Slastyona has.

The next n lines contain three integers each: l_i, r_i, c_i $(0 \le l_i < r_i \le 10^9,\ 0 \le c_i \le 10^9)$, describing one caramel cloud.

The next line contains single integer m $(1 \le m \le 3 \cdot 10^5)$ – the number of seedlings. Each of the seedlings is described with one integer k_j $(1 \le k_j \le 10^9)$ – the required number of sunny minutes.

Output

For each seedling print one integer – the minimum minute Slastyona can grow it up.

Examples

input

```
3 5
1 7 1
1 6 2
1 7 1
3
7
2
5
```

output

```
12
7
10
```

input

```
3 15
1 4 17
2 8 6
4 8 9
2
5
1
```

output

```
8
1
```

input

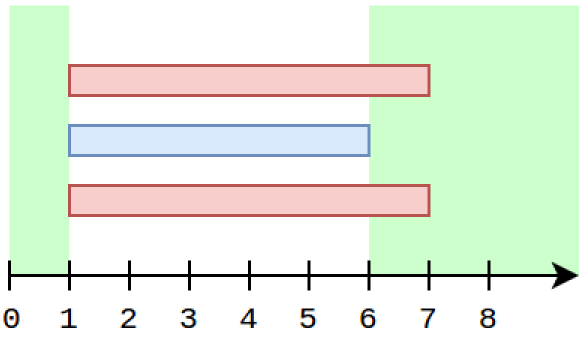
```
2 10
3 7 9
10 90 10
2
10
100
```

output

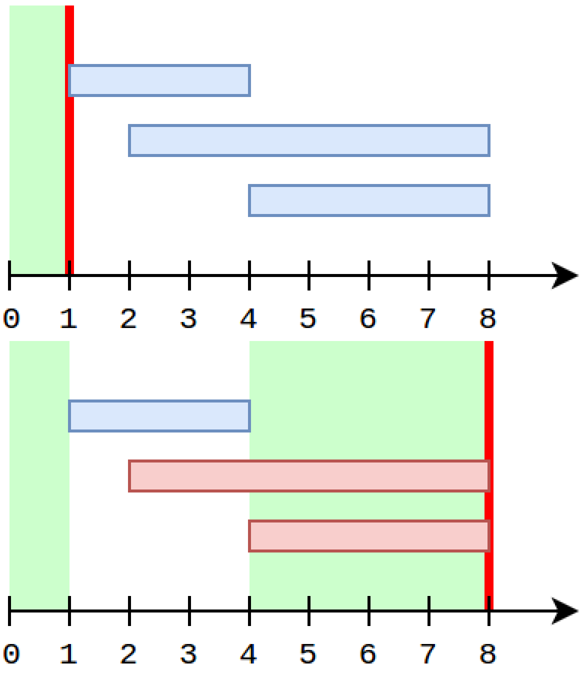
```
10
104
```

Note

Consider the first example. For each k it is optimal to dispel clouds 1 and 3. Then the remaining cloud will give shadow on time segment $[1..6]$. So, intervals $[0..1]$ and $[6..inf)$ are sunny.



In the second example for $k = 1$ it is not necessary to dispel anything, and for $k = 5$ the best strategy is to dispel clouds 2 and 3. This adds an additional sunny segment $[4..8]$, which together with $[0..1]$ allows to grow up the muffin at the eight minute.



If the third example the two seedlings are completely different. For the first one it is necessary to dispel cloud 1 and obtain a sunny segment $[0..10]$. However, the same strategy gives answer 180 for the second seedling. Instead, we can dispel cloud 2, to make segments $[0..3]$ and $[7..inf)$ sunny, and this allows up to shorten the time to 104.