A. Games on a CD

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

Several years ago Tolya had n computer games and at some point of time he decided to burn them to CD. After that he wrote down the names of the games one after another in a circle on the CD **in clockwise order**. The names were distinct, the length of each name was equal to k. The names didn't overlap.

Thus, there is a cyclic string of length $n \cdot k$ written on the CD.

Several years have passed and now Tolya can't remember which games he burned to his CD. He knows that there were g popular games that days. All of the games he burned were among these g games, and **no game was burned more than once**.

You have to restore any valid list of games Tolya could burn to the CD several years ago.

Input

The first line of the input contains two positive integers n and k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^5$) — the amount of games Tolya burned to the CD, and the length of each of the names.

The second line of the input contains one string consisting of lowercase English letters — the string Tolya wrote on the CD, split in arbitrary place. The length of the string is $n \cdot k$. It is guaranteed that the length is not greater than 10^6.

The third line of the input contains one positive integer g ($n \leq g \leq 10^5$) — the amount of popular games that could be written on the CD. It is guaranteed that the total length of names of all popular games is not greater than $2 \cdot 10^6$.

Each of the next g lines contains a single string — the name of some popular game. Each name consists of lowercase English letters and has length k. It is guaranteed that the names are distinct.

Output

If there is no answer, print "NO" (without quotes).

Otherwise, print two lines. In the first line print "YES" (without quotes). In the second line, print n integers — the games which names were written on the CD. You should print games in the order they could have been written on the CD, it means, **in clockwise order**. You can print games starting from any position. Remember, that no game was burned to the CD more than once. If there are several possible answers, print any of them.

Examples

input

```
3 1
abc
4
b
a
c
d
```

output

```
YES
2 1 3
```

input

```
4 2
aabbccdd
4
dd
ab
bc
cd
```

output

```
NO
```

B. Very simple problem

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

You are given a convex polygon. Count, please, the number of triangles that contain a given point in the plane and their vertices are the vertices of the polygon. It is guaranteed, that the point doesn't lie on the sides and the diagonals of the polygon.

Input

The first line contains integer n — the number of vertices of the polygon ($3 \le n \le 100000$). The polygon description is following: n lines containing coordinates of the vertices in clockwise order (integer x and y not greater than 10^9 by absolute value). It is guaranteed that the given polygon is nondegenerate and convex (no three points lie on the same line).

The next line contains integer t ($1 \le t \le 20$) — the number of points which you should count the answer for. It is followed by t lines with coordinates of the points (integer x and y not greater than 10^9 by absolute value).

Output

The output should contain t integer numbers, each on a separate line, where i-th number is the answer for the i-th point.

Please, do not use `%lld` specificator to read or write 64-bit integers in C++. It is preffered to use `cin` (also you may use `%I64d`).

Examples

input

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

output

```
2
```

input

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

output

```
1
0
```

input

```
5
7 6
```

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

output

```
5
3
3
4
```

C. Hot Days

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

The official capital and the cultural capital of Berland are connected by a single road running through n regions. Each region has a unique climate, so the i-th $(1 \le i \le n)$ region has a stable temperature of t_i degrees in summer.

This summer a group of m schoolchildren wants to get from the official capital to the cultural capital to visit museums and sights. The trip organizers transport the children between the cities in buses, but sometimes it is very hot. Specifically, if the bus is driving through the i-th region and has k schoolchildren, then the temperature inside the bus is $t_i + k$ degrees.

Of course, nobody likes it when the bus is hot. So, when the bus drives through the i-th region, if it has more than T_i degrees inside, each of the schoolchild in the bus demands compensation for the uncomfortable conditions. The compensation is as large as x_i rubles and it is charged in each region where the temperature in the bus exceeds the limit.

To save money, the organizers of the trip may arbitrarily add or remove extra buses in the beginning of the trip, and between regions (of course, they need at least one bus to pass any region). The organizers can also arbitrarily sort the children into buses, however, each of buses in the i-th region will cost the organizers $cost_i$ rubles. Please note that sorting children into buses takes no money.

Your task is to find the minimum number of rubles, which the organizers will have to spend to transport all schoolchildren.

Input

The first input line contains two integers n and m $(1 \le n \le 10^5; 1 \le m \le 10^6)$ — the number of regions on the way and the number of schoolchildren in the group, correspondingly. Next n lines contain four integers each: the i-th line contains t_i, T_i, x_i and $cost_i$ $(1 \le t_i, T_i, x_i, cost_i \le 10^6)$. The numbers in the lines are separated by single spaces.

Output

Print the only integer — the minimum number of roubles the organizers will have to spend to transport all schoolchildren.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use cin, cout streams or the `%I64d` specifier.

Examples

input

```
2 10
30 35 1 100
20 35 10 10
```

output

```
120
```

input

```
3 100
10 30 1000 1
```

```
5 10 1000 3
10 40 1000 100000
```

output

```
200065
```

Note

In the first sample the organizers will use only one bus to travel through the first region. However, the temperature in the bus will equal $30 + 10 = 40$ degrees and each of 10 schoolchildren will ask for compensation. Only one bus will transport the group through the second region too, but the temperature inside won't exceed the limit. Overall, the organizers will spend $100 + 10 + 10 = 120$ rubles.

D. Swaps in Permutation

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a permutation of the numbers $1, 2, ..., n$ and m pairs of positions (a_j, b_j).

At each step you can choose a pair from the given positions and swap the numbers in that positions. What is the lexicographically maximal permutation one can get?

Let p and q be two permutations of the numbers $1, 2, ..., n$. p is lexicographically smaller than the q if a number $1 \le i \le n$ exists, so $p_k = q_k$ for $1 \le k < i$ and $p_i < q_i$.

Input

The first line contains two integers n and m $(1 \le n, m \le 10^6)$ — the length of the permutation p and the number of pairs of positions.

The second line contains n distinct integers p_i $(1 \le p_i \le n)$ — the elements of the permutation p.

Each of the last m lines contains two integers (a_j, b_j) $(1 \le a_j, b_j \le n)$ — the pairs of positions to swap. Note that you are given a `positions`, not the values to swap.

Output

Print the only line with n distinct integers p'_i $(1 \le p'_i \le n)$ — the lexicographically maximal permutation one can get.

Example

input

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

output

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

E. Array and Operations

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

You have written on a piece of paper an array of n positive integers $a[1], a[2], ..., a[n]$ and m *good* pairs of integers $(i_1, j_1), (i_2, j_2), ..., (i_m, j_m)$. Each *good* pair (i_k, j_k) meets the following conditions: $i_k + j_k$ is an odd number and $1 \le i_k < j_k \le n$.

In one operation you can perform a sequence of actions:

- take one of the *good* pairs (i_k, j_k) and some integer v ($v > 1$), which divides both numbers $a[i_k]$ and $a[j_k]$;
- divide both numbers by v, i. e. perform the assignments: $a[i_k] = \frac{a[i_k]}{v}$ and $a[j_k] = \frac{a[j_k]}{v}$.

Determine the maximum number of operations you can sequentially perform on the given array. Note that one pair may be used several times in the described operations.

Input

The first line contains two space-separated integers n, m $(2 \le n \le 100, 1 \le m \le 100)$.

The second line contains n space-separated integers $a[1], a[2], ..., a[n]$ $(1 \le a[i] \le 10^9)$ — the description of the array.

The following m lines contain the description of *good* pairs. The k-th line contains two space-separated integers i_k, j_k $(1 \le i_k < j_k \le n, i_k + j_k$ is an odd number).

It is guaranteed that all the *good* pairs are distinct.

Output

Output the answer for the problem.

Examples

input

```
3 2
8 3 8
1 2
2 3
```

output

```
0
```

input

```
3 2
8 12 8
1 2
2 3
```

output

```
2
```

F. Upgrading Tree

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

You are given a tree with n vertices and you are allowed to perform **no more than** $2n$ transformations on it. Transformation is defined by three vertices x, y, y' and consists of deleting edge (x, y) and adding edge (x, y'). Transformation x, y, y' could be performed if all the following conditions are satisfied:

1. There is an edge (x, y) in the current tree.
2. After the transformation the graph remains a tree.
3. After the deletion of edge (x, y) the tree would consist of two connected components. Let's denote the set of nodes in the component containing vertex x by V_x, and the set of nodes in the component containing vertex y by V_y. Then condition $|V_x| > |V_y|$ should be satisfied, i.e. the size of the component with x should be strictly larger than the size of the component with y.

You should **minimize** the sum of squared distances between all pairs of vertices in a tree, which you could get after no more than $2n$ transformations and output any sequence of transformations leading initial tree to such state.

Note that you don't need to minimize the number of operations. It is necessary to minimize only the sum of the squared distances.

Input

The first line of input contains integer n ($1 \le n \le 2 \cdot 10^5$) — number of vertices in tree.

The next $n - 1$ lines of input contains integers a and b ($1 \le a, b \le n$, $a \ne b$) — the descriptions of edges. It is guaranteed that the given edges form a tree.

Output

In the first line output integer k ($0 \le k \le 2n$) — the number of transformations from your example, **minimizing** sum of squared distances between all pairs of vertices.

In each of the next k lines output three integers x, y, y' — indices of vertices from the corresponding transformation.

Transformations with $y = y'$ are allowed (even though they don't change tree) if transformation conditions are satisfied.

If there are several possible answers, print any of them.

Examples

input

```
3
3 2
1 3
```

output

```
0
```

input

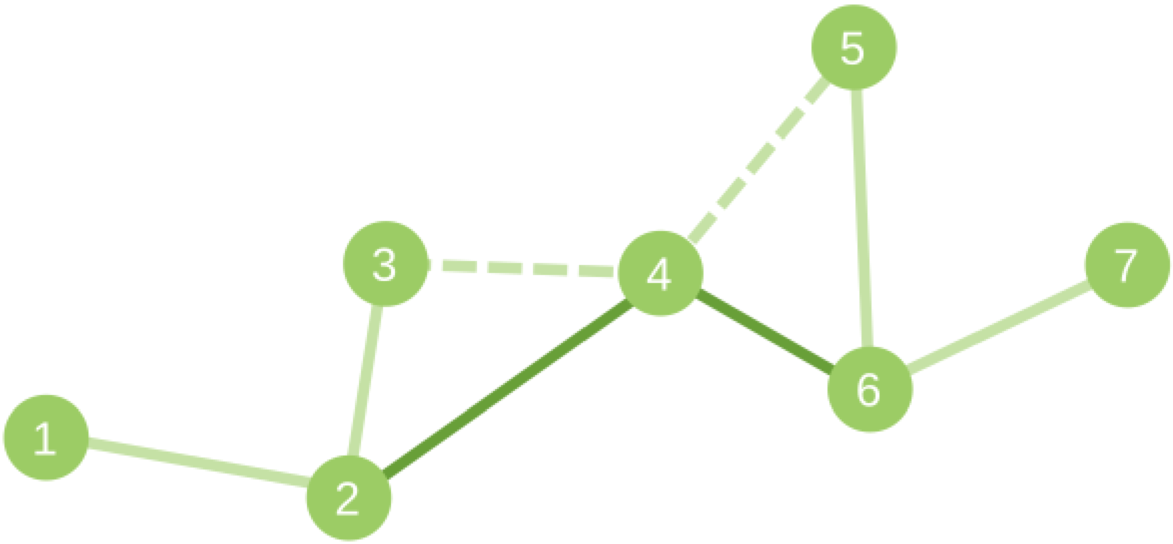
```
7
1 2
2 3
3 4
4 5
5 6
6 7
```

output

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

Note

This is a picture for the second sample. Added edges are dark, deleted edges are dotted.



G. Vasiliy's Multiset

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Author has gone out of the stories about Vasiliy, so here is just a formal task description.

You are given q queries and a multiset A, initially containing only integer 0. There are three types of queries:

1. "+ x" — add integer x to multiset A.
2. "- x" — erase one occurrence of integer x from multiset A. It's guaranteed that at least one x is present in the multiset A before this query.
3. "? x" — you are given integer x and need to compute the value $\max_{y \in A}(x \oplus y)$, i.e. the maximum value of bitwise exclusive OR (also know as XOR) of integer x and some integer y from the multiset A.

Multiset is a set, where equal elements are allowed.

Input

The first line of the input contains a single integer q ($1 \leq q \leq 200\,000$) — the number of queries Vasiliy has to perform.

Each of the following q lines of the input contains one of three characters '+', '-' or '?' and an integer x_i ($1 \leq x_i \leq 10^9$). It's guaranteed that there is at least one query of the third type.

Note, that the integer 0 will always be present in the set A.

Output

For each query of the type '?' print one integer — the maximum value of bitwise exclusive OR (XOR) of integer x_i and some integer from the multiset A.

Example

input

```
10
+ 8
+ 9
+ 11
+ 6
+ 1
? 3
- 8
? 3
? 8
? 11
```

output

```
11
10
14
13
```

Note

After first five operations multiset A contains integers 0, 8, 9, 11, 6 and 1.

The answer for the sixth query is integer $11 = 3 \oplus 8$ — maximum among integers $3 \oplus 0 = 3$, $3 \oplus 9 = 10$, $3 \oplus 11 = 8$, $3 \oplus 6 = 5$ and $3 \oplus 1 = 2$.

H. Inverse Coloring

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

You are given a square board, consisting of n rows and n columns. Each tile in it should be colored either white or black.

Let's call some coloring *beautiful* if each pair of adjacent rows are either the same or different in every position. The same condition should be held for the columns as well.

Let's call some coloring *suitable* if it is *beautiful* and there is no **rectangle** of the single color, consisting of at least k tiles.

Your task is to count the number of *suitable* colorings of the board of the given size.

Since the answer can be very large, print it modulo 998244353.

Input

A single line contains two integers n and k ($1 \le n \le 500$, $1 \le k \le n^2$) — the number of rows and columns of the board and the maximum number of tiles inside the rectangle of the single color, respectively.

Output

Print a single integer — the number of *suitable* colorings of the board of the given size modulo 998244353.

Examples

input
```
1 1
```
output
```
0
```

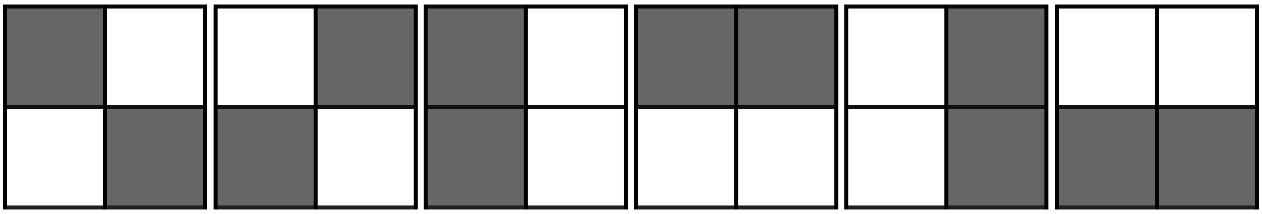
input
```
2 3
```
output
```
6
```

input
```
49 1808
```
output
```
359087121
```

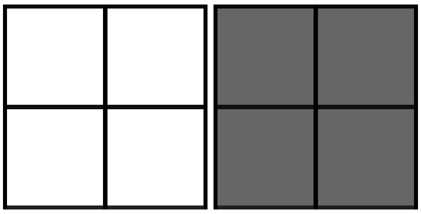
Note

Board of size 1×1 is either a single black tile or a single white tile. Both of them include a rectangle of a single color, consisting of 1 tile.

Here are the *beautiful* colorings of a board of size 2×2 that don't include rectangles of a single color, consisting of at least 3 tiles:



The rest of *beautiful* colorings of a board of size 2×2 are the following:

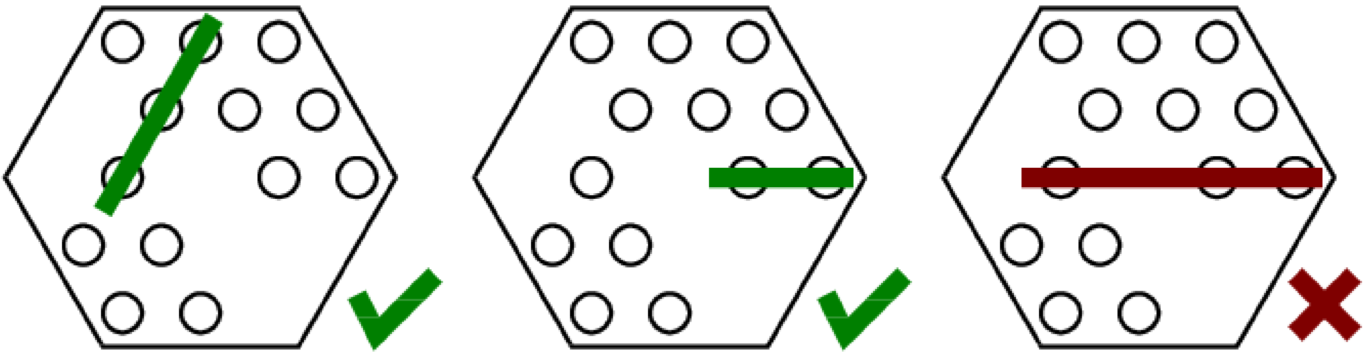


I. Sweets Game

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

Karlsson has visited Lillebror again. They found a box of chocolates and a big whipped cream cake at Lillebror's place. Karlsson immediately suggested to divide the sweets fairly between Lillebror and himself. Specifically, to play together a game he has just invented with the chocolates. The winner will get the cake as a reward.

The box of chocolates has the form of a hexagon. It contains 19 cells for the chocolates, some of which contain a chocolate. The players move in turns. During one move it is allowed to eat one or several chocolates that lay in the neighboring cells on one line, parallel to one of the box's sides. The picture below shows the examples of allowed moves and of an unacceptable one. The player who cannot make a move loses.



Karlsson makes the first move as he is Lillebror's guest and not vice versa. The players play optimally. Determine who will get the cake.

Input

The input data contains 5 lines, containing 19 words consisting of one symbol. The word "O" means that the cell contains a chocolate and a "." stands for an empty cell. It is guaranteed that the box contains at least one chocolate. See the examples for better understanding.

Output

If Karlsson gets the cake, print "Karlsson" (without the quotes), otherwise print "Lillebror" (yet again without the quotes).

Examples

input

```
  . . .
 . . O .
. . O O .
 . . . .
  . . .
```

output

```
Lillebror
```

input

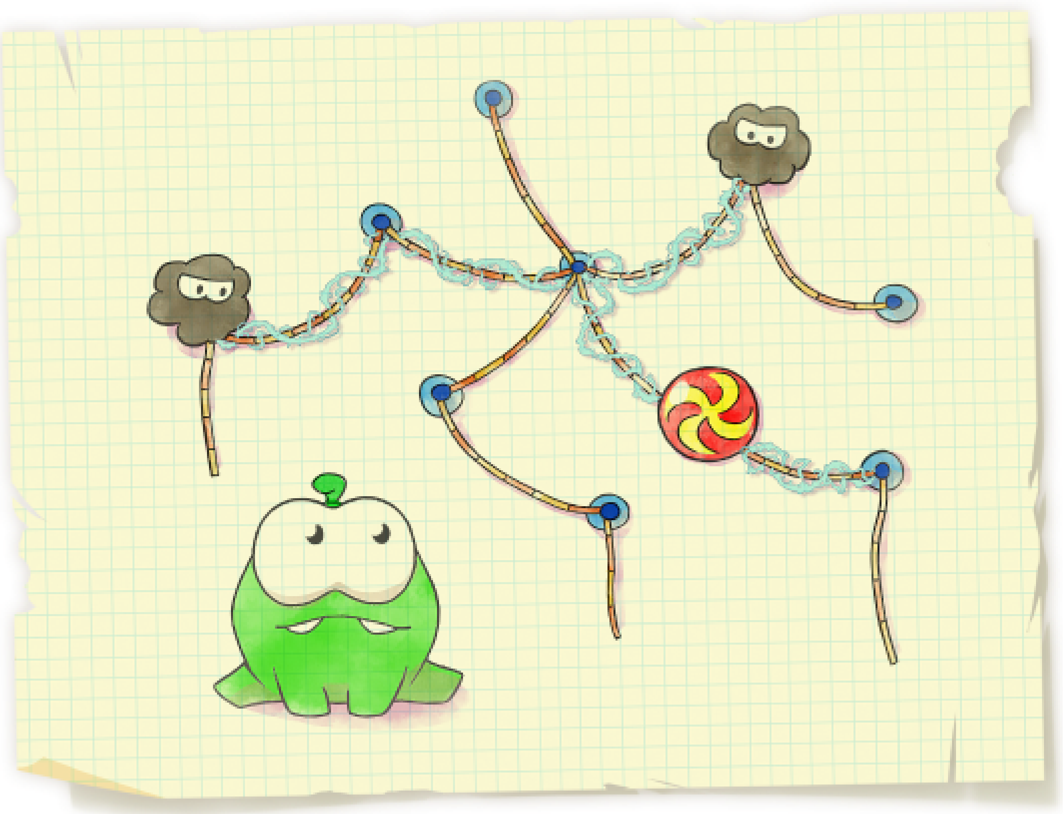
```
  . . .
 . . . O
. . . O .
 O . O .
  . O .
```

output
Karlsson

J. Spiders Evil Plan

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

Spiders are Om Nom's old enemies. They love eating candies as much as he does and that's why they keep trying to keep the monster away from his favorite candies. They came up with an evil plan to trap Om Nom.



Let's consider a rope structure consisting of n nodes and $n - 1$ ropes connecting the nodes. The structure is connected, thus, the ropes and the nodes form a tree. Each rope of the formed structure is associated with its length. A candy is tied to node x of the structure. Om Nom really wants to eat this candy.

The y spiders are trying to stop him from doing it. They decided to entangle the candy and some part of the structure into a web, thus attaching the candy to as large as possible part of the rope structure.

Each spider can use his web to cover all ropes on the path between two arbitrary nodes a and b. Thus, y spiders can cover the set of ropes which is a union of y paths in the given tree. These y paths can arbitrarily intersect each other. The spiders want the following conditions to be hold:

- the node containing the candy is adjacent to at least one rope covered with a web
- the ropes covered with the web form a connected structure (what's the idea of covering with a web the ropes that are not connected with the candy?)
- the total length of the ropes covered with web is as large as possible

The spiders haven't yet decided to what node of the structure they will tie the candy and how many spiders will cover the structure with web, so they asked you to help them. Help them calculate the optimal plan for multiple values of x and y.

Input

The first line contains numbers n and q ($1 \le n, q \le 10^5$) — the number of nodes in the structure and the number of questions that the spiders want to ask you.

The next $n - 1$ lines determine the rope structure. The i-th line contains three integers u_i, v_i, l_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$, $1 \le l_i \le 1000$), showing that there is a rope of length l_i between nodes u_i and v_i.

Next q lines describe the spiders' questions. As they want you to answer their question online, they encoded their messages in a special manner.

Each of the next q lines contains two numbers x_i, y_i. In the first question of the spiders $x = x_1, y = y_1$.

To calculate values x and y in the spiders' i-th ($2 \le i \le q$) question, you need to use the following formulas:

$$x = ((x_i + Ans_{i-1} - 1) \bmod n) + 1$$

$$y = ((y_i + Ans_{i-1} - 1) \bmod n) + 1$$

where Ans_{i-1} is the total length of the ropes covered by a web in the answer for the $(i - 1)$-th question.

The following inequality holds: $1 \le x_i, y_i \le n$.

Output

For each question of the spiders print on a separate line a single integer Ans_i — the total length of the ropes covered with web in the optimal plan.

Examples

input

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

output

```
14
13
17
```

K. A Simple Task

time limit per test: 5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

This task is very simple. Given a string S of length n and q queries each query is on the format $i\ j\ k$ which means sort the substring consisting of the characters from i to j in non-decreasing order if $k = 1$ or in non-increasing order if $k = 0$.

Output the final string after applying the queries.

Input

The first line will contain two integers n, q ($1 \le n \le 10^5$, $0 \le q \le 50\,000$), the length of the string and the number of queries respectively.

Next line contains a string S itself. It contains only lowercase English letters.

Next q lines will contain three integers each i, j, k ($1 \le i \le j \le n$, $k \in \{0, 1\}$).

Output

Output one line, the string S after applying the queries.

Examples

input

```
10 5
abacdabcda
7 10 0
5 8 1
1 4 0
3 6 0
7 10 1
```

output

```
cbcaaaabdd
```

input

```
10 1
agjucbvdfk
1 10 1
```

output

```
abcdfgjkuv
```

Note

First sample test explanation:

$$abacda\mathbf{bcda} \to abacda\mathbf{dcba}$$

$$abac\mathbf{dadc}ba \to abac\mathbf{acdd}ba$$

$$\mathbf{abac}acddba \to \mathbf{cbaa}acddba$$

$$cb\mathbf{aaac}ddba \to cb\mathbf{caaa}ddba$$

$$cbcaaa\mathbf{ddba} \to cbcaaa\mathbf{abdd}$$

L. Plane of Tanks: Duel

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

Vasya plays the Plane of Tanks.

Tanks are described with the following attributes:

- the number of hit points;
- the interval between two gun shots (the time required to recharge the gun);
- the probability that the gun shot will not pierce armor of the enemy tank;
- the damage to the enemy's tank.

The gun damage is described with a segment $[l, r]$, where l and r are integer numbers. The potential gun damage x is chosen with equal probability among all integer numbers of the segment $[l, r]$. If the shot pierces the armor of an enemy's tank then the enemy loses x hit points. If the number of hit points becomes non-positive then the enemy tank is considered destroyed.

It is possible that the shot does not pierce the armor of a tank. In this case the number of hit points doesn't change. The probability that the armor will not be pierced is considered as the shooting tank attribute and does not depend on players' behavior.

The victory is near and there is only one enemy tank left. Vasya is ready for the battle — one more battle between the Good and the Evil is inevitable! Two enemies saw each other and each of them fired a shot at the same moment... The last battle has begun! Help Vasya to determine what is the probability that he will win the battle by destroying the enemy tank?

If both tanks are destroyed (after simultaneous shots), then Vasya is considered a winner. You can assume that each player fires a shot just after the gun recharge and each tank has infinite number of ammo.

Input

The first line contains five integer numbers separated with spaces describing Vasya's tank: the number of hit points hp $(10 \le hp \le 200)$, the interval between two shots dt $(1 \le dt \le 30)$, gun damage segment l and r $(10 \le l \le r \le 100)$, the probability that the enemy's tank armor will not be pierced p $(0 \le p \le 100)$ (percents).

The second line describes the tank of Vasya's enemy in the same format.

Output

Print the only number with absolute or relative error no more than 10^{-4} — probability of Vasya's victory.

Examples

input

```
100 3 50 50 0
100 3 50 50 0
```

output

```
1.000000
```

input

```
100 3 50 50 0
100 2 48 50 0
```

output

```
0.888889
```

input

```
100 3 50 50 0
100 1 50 50 50
```

output

```
0.500000
```

Note

In the first example both tanks are destroyed at once after the second shot. The probability of destroying the enemy tank is 1.

In the second example Vasya's enemy tank fires the second shot before Vasya's tank, but has no time for the third shot. In order to destroy Vasya's tank it is necessary to fire two shots with damage 50. The probability of that event is $\left(\frac{1}{3}\right)^2 = \frac{1}{9}$. Otherwise, Vasya wins.

In the third example Vasya's enemy tank fires three shots with probability of armor piercing 0.5. In order to destroy Vasya's tank it is necessary that at least 2 of 3 shots pierce the armor of Vasya's tank. The probability of this event is 0.5.

M. Timofey and our friends animals

time limit per test: 7 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

After his birthday party, Timofey went to his favorite tree alley in a park. He wants to feed there his favorite birds — crows.

It's widely known that each tree is occupied by a single crow family. The trees in the alley form a row and are numbered from 1 to n. Some families are friends to each other. For some reasons, two families can be friends only if they live not too far from each other, more precisely, there is no more than $k - 1$ trees between any pair of friend families. Formally, the family on the u-th tree and the family on the v-th tree can be friends only if $|u - v| \le k$ holds.

One of the friendship features is that if some family learns that Timofey is feeding crows somewhere, it notifies about this all friend families. Thus, after Timofey starts to feed crows under some tree, all the families that are friends to the family living on this tree, as well as their friends and so on, fly to the feeding place. Of course, the family living on the tree also comes to the feeding place.

Today Timofey came to the alley and noticed that all the families that live on trees with numbers strictly less than l or strictly greater than r have flown away. Thus, it is not possible to pass the information about feeding through them. Moreover, there is no need to feed them. Help Timofey to learn what is the minimum number of trees under which he has to feed crows so that all the families that have remained will get the information about feeding. You are given several situations, described by integers l and r, you need to calculate the answer for all of them.

Input

The first line contains integers n and k ($1 \le n \le 10^5$, $1 \le k \le 5$), where n is the number of trees, and k is the maximum possible distance between friend families.

The next line contains single integer m ($0 \le m \le n \cdot k$) — the number of pair of friend families.

Each of the next m lines contains two integers u and v ($1 \le u, v \le 10^5$), that means that the families on trees u and v are friends. It is guaranteed that $u \ne v$ and $|u - v| \le k$. All the given pairs are distinct.

The next line contains single integer q ($1 \le q \le 10^5$) — the number of situations you need to calculate the answer in.

Each of the next q lines contains two integers l and r ($1 \le l \le r \le 10^5$), that means that in this situation families that have flown away lived on such trees x, so that either $x < l$ or $x > r$.

Output

Print q lines. Line i should contain single integer — the answer in the i-th situation.

Example

input

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

```
1 3
1 5
```

output

```
1
2
1
1
2
```

Note

In the first example the following family pairs are friends: $(1, 3)$, $(2, 3)$ and $(4, 5)$.

- In the first situation only the first family has remained, so the answer is 1.
- In the second situation the first two families have remained, and they aren't friends, so the answer is 2.
- In the third situation the families 2 and 3 are friends, so it is enough to feed any of them, the answer is 1.
- In the fourth situation we can feed the first family, then the third family will get the information from the first family, and the second family will get the information from the third. The answer is 1.
- In the fifth situation we can feed the first and the fifth families, so the answer is 2.