A1. Heidi Learns Hashing (Easy)

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

Melody Pond was stolen from her parents as a newborn baby by Madame Kovarian, to become a weapon of the Silence in their crusade against the Doctor. Madame Kovarian changed Melody's name to River Song, giving her a new identity that allowed her to kill the Eleventh Doctor.

Heidi figured out that Madame Kovarian uses a very complicated hashing function in order to change the names of the babies she steals. In order to prevent this from happening to future Doctors, Heidi decided to prepare herself by learning some basic hashing techniques.

The first hashing function she designed is as follows.

Given two positive integers (x, y) she defines $H(x,y) := x^2 + 2xy + x + 1$.

Now, Heidi wonders if the function is reversible. That is, given a positive integer r, can you find a pair (x, y) (of positive integers) such that $H(x, y) = r$?

If multiple such pairs exist, output the one with smallest possible x. If there is no such pair, output "NO".

Input

The first and only line contains an integer r ($1 \le r \le 10^{12}$).

Output

Output integers x, y such that $H(x,y) = r$ and x is smallest possible, or "NO" if no such pair exists.

Examples

input

```
19
```

output

```
1 8
```

input

```
16
```

output

```
NO
```

A2. Heidi Learns Hashing (Medium)

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

After learning about polynomial hashing, Heidi decided to learn about shift-xor hashing. In particular, she came across this interesting problem.

Given a bitstring $y \in \{0,1\}^n$ find out the number of different k ($0 \le k < n$) such that there exists $x \in \{0,1\}^n$ for which $y = x \oplus \mathrm{shift}^k(x)$.

In the above, $\oplus$ is the xor operation and shift^k is the operation of shifting a bitstring cyclically to the right k times. For example, $001 \oplus 111 = 110$ and $\mathrm{shift}^3(00010010111000) = 00000010010111$.

Input

The first line contains an integer n ($1 \le n \le 2 \cdot 10^5$), the length of the bitstring y.

The second line contains the bitstring y.

Output

Output a single integer: the number of suitable values of k.

Example

input
4 1010

output
3

Note

In the first example:

- $1100 \oplus \text{shift}^1(1100) = 1010$
- $1000 \oplus \text{shift}^2(1000) = 1010$
- $0110 \oplus \text{shift}^3(0110) = 1010$

There is no x such that $x \oplus x = 1010$, hence the answer is 3.

A3. Heidi Learns Hashing (Hard)

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

Now Heidi is ready to crack Madame Kovarian's hashing function.

Madame Kovarian has a very strict set of rules for name changes. Two names can be interchanged only if using the following hashing function on them results in a collision. However, the hashing function is parametrized, so one can always find a set of parameters that causes such a collision. Heidi decided to exploit this to her advantage.

Given two strings w_1, w_2 of equal length n consisting of lowercase English letters and an integer m.

Consider the standard polynomial hashing function:

$$H_p(w) := \left(\sum_{i=0}^{|w|-1} w_i r^i\right) \text{mod}(p)$$

where p is some prime, and r is some number such that $2 \leq r \leq p-2$.

The goal is to find r and a prime p ($m \leq p \leq 10^9$) such that $H_p(w_1) = H_p(w_2)$.

Strings w_1 and w_2 are sampled independently at random from all strings of length n over lowercase English letters.

Input

The first line contains two integers n and m ($10 \leq n \leq 10^5$, $2 \leq m \leq 10^5$).

The second and the third line, respectively, contain the words w_1, w_2 that were sampled independently at random from all strings of length n over lowercase English letters.

Output

Output integers p, r.

p should be a prime in the range $[m, 10^9]$ and r should be an integer satisfying $r \in [2, p-2]$.

At least one solution is guaranteed to exist. In case multiple solutions exist, print any of them.

Examples

input
10 5 bgcbaaaaaa cccaaaaaaa

output
5 2

input
10 100 melodypond riversongg

output
118219 79724

Note

In the first example, note that even though $p = 3$ and $r = 2$ also causes a colision of hashes, it is not a correct solution, since m is 5 and thus we want $p \geq 5$.

In the second example, we are aware of the extra 'g' at the end. We just didn't realize that "River Song" and "Melody Pond" have different lengths...

B1. The Doctor Meets Vader (Easy)

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

Heidi and Doctor Who hopped out of the TARDIS and found themselves at EPFL in 2018. They were surrounded by stormtroopers and Darth Vader was approaching. Miraculously, they managed to escape to a nearby rebel base but the Doctor was very confused. Heidi reminded him that last year's HC2 theme was Star Wars. Now he understood, and he's ready to face the evils of the Empire!

The rebels have s spaceships, each with a certain attacking power a.

They want to send their spaceships to destroy the empire bases and steal enough gold and supplies in order to keep the rebellion alive.

The empire has b bases, each with a certain defensive power d, and a certain amount of gold g.

A spaceship can attack all the bases which have a defensive power less than or equal to its attacking power.

If a spaceship attacks a base, it steals all the gold in that base.

The rebels are still undecided which spaceship to send out first, so they asked for the Doctor's help. They would like to know, for each spaceship, the maximum amount of gold it can steal.

Input

The first line contains integers s and b ($1 \leq s, b \leq 10^5$), the number of spaceships and the number of bases, respectively.

The second line contains s integers a ($0 \leq a \leq 10^9$), the attacking power of each spaceship.

The next b lines contain integers d, g ($0 \leq d \leq 10^9$, $0 \leq g \leq 10^4$), the defensive power and the gold of each base, respectively.

Output

Print s integers, the maximum amount of gold each spaceship can steal, in the same order as the spaceships are given in the input.

Example

input

```
5 4
1 3 5 2 4
0 1
4 2
2 8
9 4
```

output

```
1 9 11 9 11
```

Note

The first spaceship can only attack the first base.

The second spaceship can attack the first and third bases.

The third spaceship can attack the first, second and third bases.

B2. The Doctor Meets Vader (Medium)

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

Thanks to the Doctor's help, the rebels managed to steal enough gold to launch a full-scale attack on the Empire! However, Darth Vader is looking for revenge and wants to take back his gold.

The rebels have hidden the gold in various bases throughout the galaxy. Darth Vader and the Empire are looking to send out their spaceships to attack these bases.

The galaxy can be represented as an undirected graph with n planets (nodes) and m wormholes (edges), each connecting two planets.

A total of s empire spaceships and b rebel bases are located at different planets in the galaxy.

Each spaceship is given a location x, denoting the index of the planet on which it is located, an attacking strength a, and a certain amount of fuel f.

Each base is given a location x, and a defensive strength d.

A spaceship can attack a base if both of these conditions hold:

- the spaceship's attacking strength is greater or equal than the defensive strength of the base
- the spaceship's fuel is greater or equal to the shortest distance, computed as the number of wormholes, between the spaceship's planet and the base's planet

Vader is very particular about his attacking formations. He requires that each spaceship is to attack at most one base and that each base is to be attacked by at most one spaceship.

Vader knows that the rebels have hidden k gold in each base, so he will assign the spaceships to attack bases in such a way that maximizes the number of bases attacked.

Therefore, for each base that is attacked, the rebels lose k gold.

However, the rebels have the ability to create any number of dummy bases. With the Doctor's help, these bases would exist beyond space and time, so all spaceship can reach them and attack them. Moreover, a dummy base is designed to seem irresistible: that is, it will always be attacked by some spaceship.

Of course, dummy bases do not contain any gold, but creating such a dummy base costs h gold.

What is the minimum gold the rebels can lose if they create an optimal number of dummy bases?

Input

The first line contains two integers n and m ($1 \le n \le 100$, $0 \le m \le 10000$), the number of nodes and the number of edges, respectively.

The next m lines contain two integers u and v ($1 \le u$, $v \le n$) denoting an undirected edge between the two nodes.

The next line contains four integers s, b, k and h ($1 \le s$, $b \le 1000$, $0 \le k$, $h \le 10^9$), the number of spaceships, the number of bases, the cost of having a base attacked, and the cost of creating a dummy base, respectively.

The next s lines contain three integers x, a, f ($1 \le x \le n$, $0 \le a$, $f \le 10^9$), denoting the location, attack, and fuel of the spaceship.

The next b lines contain two integers x, d ($1 \le x \le n$, $0 \le d \le 10^9$), denoting the location and defence of the base.

Output

Print a single integer, the minimum cost in terms of gold.

Example

input

```
6 7
1 2
2 3
3 4
4 6
6 5
4 4
3 6
4 2 7 3
1 10 2
3 8 2
5 1 0
6 5 4
3 7
5 2
```

output

```
12
```

Note

One way to minimize the cost is to build 4 dummy bases, for a total cost of $4 \times 3 = 12$.

One empire spaceship will be assigned to attack each of these dummy bases, resulting in zero actual bases attacked.

B3. The Doctor Meets Vader (Hard)

time limit per test: 4 seconds

memory limit per test: 512 megabytes

input: standard input

output: standard output

The rebels have saved enough gold to launch a full-scale attack. Now the situation is flipped, the rebels will send out the spaceships to attack the Empire bases!

The galaxy can be represented as an undirected graph with n planets (nodes) and m wormholes (edges), each connecting two planets.

A total of s rebel spaceships and b empire bases are located at different planets in the galaxy.

Each spaceship is given a location x, denoting the index of the planet on which it is located, an attacking strength a, a certain amount of fuel f, and a price to operate p.

Each base is given a location x, a defensive strength d, and a certain amount of gold g.

A spaceship can attack a base if both of these conditions hold:

- the spaceship's attacking strength is greater or equal than the defensive strength of the base
- the spaceship's fuel is greater or equal to the shortest distance, computed as the number of wormholes, between the spaceship's node and the base's node

The rebels are very proud fighters. So, if a spaceship cannot attack any base, no rebel pilot will accept to operate it.

If a spaceship is operated, the profit generated by that spaceship is equal to the gold of the base it attacks minus the price to operate the spaceship. Note that this might be negative. A spaceship that is operated will attack the base that maximizes its profit.

Darth Vader likes to appear rich at all times. Therefore, whenever a base is attacked and its gold stolen, he makes sure to immediately refill that base with gold.

Therefore, for the purposes of the rebels, multiple spaceships can attack the same base, in which case each spaceship will still receive all the gold of that base.

The rebels have tasked Heidi and the Doctor to decide which set of spaceships to operate in order to maximize the total profit.

However, as the war has been going on for a long time, the pilots have formed unbreakable bonds, and some of them refuse to operate spaceships if their friends are not also operating spaceships.

They have a list of k dependencies of the form s_1, s_2, denoting that spaceship s_1 can be operated only if spaceship s_2 is also operated.

Input

The first line of input contains integers n and m ($1 \leq n \leq 100$, $0 \leq m \leq 10000$), the number of nodes and the number of edges, respectively.

The next m lines contain integers u and v ($1 \leq u, v \leq n$) denoting an undirected edge between the two nodes.

The next line contains integers s, b and k ($1 \leq s, b \leq 10^5$, $0 \leq k \leq 1000$), the number of spaceships, bases, and dependencies, respectively.

The next s lines contain integers x, a, f, p ($1 \leq x \leq n$, $0 \leq a, f, p \leq 10^9$), denoting the location, attack, fuel, and price of the spaceship. Ships are numbered from 1 to s.

The next b lines contain integers x, d, g ($1 \leq x \leq n$, $0 \leq d, g \leq 10^9$), denoting the location, defence, and gold of the base.

The next k lines contain integers s_1 and s_2 ($1 \leq s_1, s_2 \leq s$), denoting a dependency of s_1 on s_2.

Output

Print a single integer, the maximum total profit that can be achieved.

Example

input

```
6 7
1 2
2 3
3 4
4 6
6 5
4 4
3 6
4 2 2
1 10 2 5
3 8 2 7
5 1 0 2
6 5 4 1
3 7 6
5 2 3
4 2
3 2
```

output

```
2
```

Note

The optimal strategy is to operate spaceships 1, 2, and 4, which will attack bases 1, 1, and 2, respectively.

C1. Heidi and the Turing Test (Easy)

time limit per test: 3 seconds

memory limit per test: 256 megabytes

input: standard input
output: standard output

The Cybermen and the Daleks have long been the Doctor's main enemies. Everyone knows that both these species enjoy destroying everything they encounter. However, a little-known fact about them is that they both also love taking Turing tests!

Heidi designed a series of increasingly difficult tasks for them to spend their time on, which would allow the Doctor enough time to save innocent lives!

The funny part is that these tasks would be very easy for a human to solve.

The first task is as follows. There are some points on the plane. All but one of them are on the boundary of an axis-aligned square (its sides are parallel to the axes). Identify that point.

Input

The first line contains an integer n ($2 \le n \le 10$).

Each of the following $4n+1$ lines contains two integers x_i, y_i ($0 \le x_i, y_i \le 50$), describing the coordinates of the next point.

It is guaranteed that there are at least n points on each side of the square and all $4n+1$ points are distinct.

Output

Print two integers — the coordinates of the point that is not on the boundary of the square.

Examples

input

```
2
0 0
0 1
0 2
1 0
1 1
1 2
2 0
2 1
2 2
```

output

```
1 1
```

input

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

output

```
0 3
```

Note

In both examples, the square has four sides $x=0$, $x=2$, $y=0$, $y=2$.

C2. Heidi and the Turing Test (Medium)

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

The Cybermen solved that first test much quicker than the Daleks. Luckily for us, the Daleks were angry (shocking!) and they destroyed some of the Cybermen.

After the fighting stopped, Heidi gave them another task to waste their time on.

There are n points on a plane. Given a radius r, find the maximum number of points that can be covered by an L^1-ball with radius r.

An L^1-ball with radius r and center (x_0, y_0) in a 2D-plane is defined as the set of points (x, y) such that the Manhattan distance between (x_0, y_0) and (x, y) is at most r.

Manhattan distance between (x_0, y_0) and (x, y) is defined as $|x - x_0| + |y - y_0|$.

Input

The first line contains two integers n, r ($1 \le n \le 300\,000, 1 \le r \le 10^6$), the number of points and the radius of the ball, respectively.

Each of the next n lines contains integers x_i, y_i ($-10^6 \leq x_i, y_i \leq 10^6$), describing the coordinates of the i-th point.

It is guaranteed, that all points are distinct.

Output

Print one integer — the maximum number points that an L^1-ball with radius r can cover.

Examples

input

```
5 1
1 1
1 -1
-1 1
-1 -1
2 0
```

output

```
3
```

input

```
5 2
1 1
1 -1
-1 1
-1 -1
2 0
```

output

```
5
```

Note

In the first example, a ball centered at $(1, 0)$ covers the points $(1, 1)$, $(1, -1)$, $(2, 0)$.

In the second example, a ball centered at $(0, 0)$ covers all the points.

Note that x_0 and y_0 need not be integer.

C3. Heidi and the Turing Test (Hard)

time limit per test: 15 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The Cybermen have again outwitted the Daleks! Unfortunately, this time the Daleks decided to abandon these tasks altogether, which means the Doctor has to deal with them.

The Doctor can handle the Daleks on his own, but Heidi now has to make sure that the Cybermen are kept busy with this next task.

There are k rings on a plane. For each ring, n points are uniformly sampled with a small random noise. The task is to recover the rings given only the noisy samples.

The rings and the samples are generated as follows. The center of a ring is uniformly sampled from a disk of radius $1\,000\,000$ centered at the origin, and the radius of the ring is uniformly sampled from $[250\,000, 750\,000]$. Let R be a ring with center (x, y) and radius r. To sample a point from R, an angle θ is uniformly sampled from $[0, 2\pi]$ and a distance d is uniformly sampled from $[0.9r, 1.1r]$. The coordinates of the sampled point are then $(x + d\cos(\theta), y + d\sin(\theta))$ rounded to the closest integers.

The distance between rings is measured by their Hausdorff distance. In our case, the distance between two rings R_1, R_2 can be written as follow. Let d be the distance between the two centers and r_1, r_2 be the radii. Then the distance is

$$dist(R_1, R_2) = \max(\min(d_{--}, d_{-+}), \min(d_{+-}, d_{++}), \min(d_{--}, d_{+-}), \min(d_{-+}, d_{++}))$$

, where $d_{++} = |d + r_1 + r_2|$, $d_{+-} = |d + r_1 - r_2|$, $d_{-+} = |d - r_1 + r_2|$, $d_{--} = |d - r_1 - r_2|$.

We say that a ring R_0 is recovered if one of the rings R in the output has Hausdorff distance less than $100\,000$ from R_0.

An output is accepted if all the rings are recovered. It is guaranteed that the distances between any two rings is greater than $600\,000$.

Remember that a human can very easily solve this task, so make sure that no human traitors are helping the Cybermen complete this task.

Input

The first line contains an integer k ($1 \leq k \leq 4$), the number of rings.

The second line contains an integer n ($100 \leq n \leq 1\,000$), the number of samples per ring.

The following $n \times k$ lines contain the samples, one sample per line.

Each line contains a pair of integers x_i, y_i, where (x_i, y_i) are the coordinates of the i-th sample.

Output

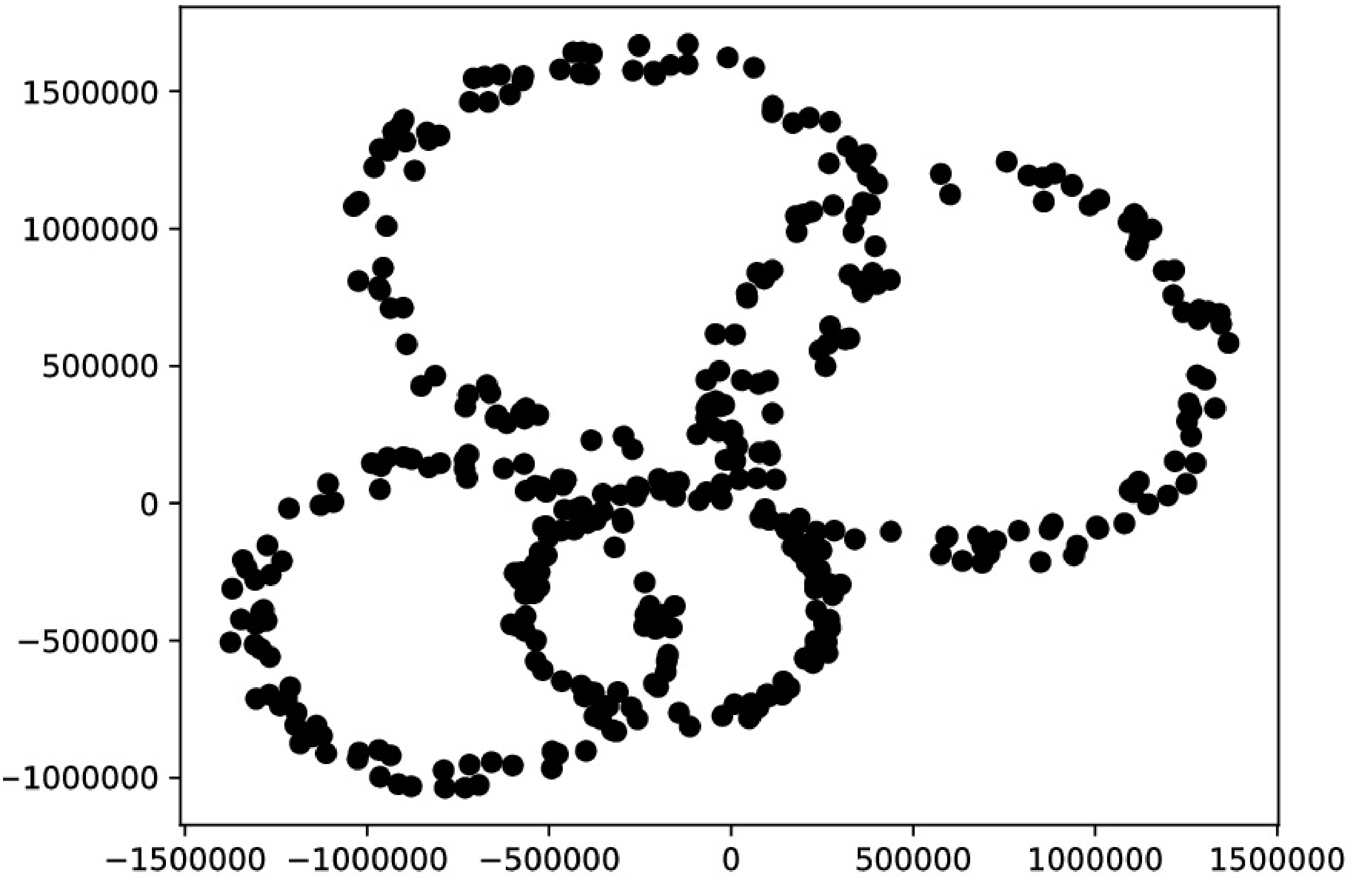
Print k lines, each describing a single ring.

For each line, print three real numbers x_i, y_i, r_i, where (x_i, y_i) and r_i are the coordinates and the radius of the i-th ring.

The order of the rings does not matter.

Note

Here is how one of tests with $k = 4$ and $n = 100$ looks like.



You can download the sample input and output here.

D1. Parallel Universes (Easy)

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

The Third Doctor Who once correctly said that travel between parallel universes is "like travelling sideways". However, he incorrectly thought that there were infinite parallel universes, whereas in fact, as we now all know, there will never be more than 250.

Heidi recently got her hands on a multiverse observation tool. She was able to see all n universes lined up in a row, with non-existent links between them. She also noticed that the Doctor was in the k-th universe.

The tool also points out that due to restrictions originating from the space-time discontinuum, the number of universes will never exceed m.

Obviously, the multiverse is unstable because of free will. Each time a decision is made, one of two events will randomly happen: a new parallel universe is created, or a non-existent link is broken.

More specifically,

- When a universe is created, it will manifest itself between any two adjacent universes or at one of the ends.
- When a link is broken, it could be cut between any two adjacent universes. After separating the multiverse into two segments, the segment *NOT* containing the Doctor will cease to exist.

Heidi wants to perform a simulation of t decisions. Each time a decision is made, Heidi wants to know the length of the multiverse (i.e. the number of universes), and the position of the Doctor.

Input

The first line contains four integers n, k, m and t ($2 \le k \le n \le m \le 250$, $1 \le t \le 1000$).

Each of the following t lines is in one of the following formats:

- "$1\ i$" — meaning that a universe is inserted at the position i ($1 \le i \le l + 1$), where l denotes the current length of the multiverse.
- "$0\ i$" — meaning that the i-th link is broken ($1 \le i \le l - 1$), where l denotes the current length of the multiverse.

Output

Output t lines. Each line should contain l, the current length of the multiverse and k, the current position of the Doctor.

It is guaranteed that the sequence of the steps will be valid, i.e. the multiverse will have length at most m and when the link breaking is performed, there will be at least one universe in the multiverse.

Example

input

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

output

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

Note

The multiverse initially consisted of 5 universes, with the Doctor being in the second.

First, link 1 was broken, leaving the multiverse with 4 universes, and the Doctor in the first.

Then, a universe was added to the leftmost end of the multiverse, increasing the multiverse length to 5, and the Doctor was then in the second universe.

Then, the rightmost link was broken.

Finally, a universe was added between the first and the second universe.

D2. Parallel Universes (Hard)

time limit per test: 4 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Heidi enjoyed performing the simulation because she knew exactly when a new universe would be formed and where, and when a non-existent link would be broken and where.

However, the multiverse itself works in mysterious ways. Well, it works using probabilities, which to some people is mysterious.

At each unit time, when a decision is made, one of the two events will happen randomly. Let's denote l as the current length of the multiverse. With a probability of $p_{create} = 1 - \frac{l}{m}$, a universe will be created. With a probability of $p_{break} = \frac{l}{m}$, a non-existent link will be broken at some position.

More specifically,

- When a universe is created, it will manifest itself between any two adjacent universes or at one of the ends. Each position occurs with a probability of $\frac{1}{l+1}$.
- When a link is broken, it could be cut between any two adjacent universes, each with a probability of $\frac{1}{l-1}$. After separating the multiverse into two segments, the segment *NOT* containing the Doctor will cease to exist.

As earlier, the Doctor remains in the same universe. However, if at some point the multiverse breaks in such a way that the Doctor finds himself at the leftmost or rightmost end of it, the TARDIS stops functioning.

In such a case, the Doctor must actually walk across the multiverse to find the tools to fix it.

We are interested in the expected value of the length of the multiverse when such an event occurs.

Input

The first and only line contains three integers n, k and m $(1 \le k \le n \le m \le 250)$, the initial length of the multiverse, the initial position of the Doctor, and the maximum possible length of the multiverse.

Output

Output a single integer on a single line, indicating the expected length of the multiverse.

If the answer is $\frac{p}{q}$, please print r where $p \equiv r \cdot q (\mod 10^9 + 7)$.

Examples

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

input
```
2 2 10
```
output
```
2
```

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

input
```
3 2 5
```
output
```
941659828
```

input
```
10 4 20
```
output
```
196683114
```

Note

For the first and the second test case, without any change to the multiverse, the Doctor is already at one of the ends.

For the third test case, the multiverse can only break at a position, which renders the Doctor at one of its ends.

For the fourth case, things seem to be a little more complicated, because the multiverse can grow and then be broken again.

E1. Daleks' Invasion (easy)

time limit per test: 6 seconds
memory limit per test: 128 megabytes
input: standard input
output: standard output

Heidi found out that the Daleks have created a network of bidirectional Time Corridors connecting different destinations (at different times!). She suspects that they are planning another invasion on the entire Space and Time. In order to counter the invasion, she plans to deploy a trap in the Time Vortex, along a carefully chosen Time Corridor. She knows that tinkering with the Time Vortex is dangerous, so she consulted the Doctor on how to proceed. She has learned the following:

- Different Time Corridors require different amounts of energy to keep stable.
- Daleks are unlikely to use all corridors in their invasion. They will pick a set of Corridors that requires the smallest total energy to maintain, yet still makes (time) travel possible between any two destinations (for those in the know: they will use a minimum spanning tree).
- Setting the trap may modify the energy required to keep the Corridor stable.

Heidi decided to carry out a field test and deploy one trap, placing it along the first Corridor. But she needs to know whether the Daleks are going to use this corridor after the deployment of the trap.

She gives you a map of Time Corridors (an undirected graph) with energy requirements for each Corridor.

For a Corridor c, $E_{max}(c)$ is the largest $e \le 10^9$ such that if we changed the required amount of energy of c to e, then the Daleks may still be using c in their invasion (that is, it belongs to some minimum spanning tree). Your task is to calculate $E_{max}(c_1)$ for the Corridor c_1 that Heidi plans to arm with a trap, which is the first edge in the graph.

Input

The first line contains integers n and m ($2 \le n \le 10^5$, $n - 1 \le m \le 10^6$), number of destinations to be invaded and the number of Time Corridors.

Each of the next m lines describes a Corridor: destinations a, b and energy e ($1 \le a, b \le n$, $a \neq b$, $0 \le e \le 10^9$).

It's guaranteed, that no pair $\{a, b\}$ will repeat and that the graph is connected — that is, it is possible to travel between any two destinations using zero or more Time Corridors.

Output

Output a single integer: $E_{max}(c_1)$ for the first Corridor c_1 from the input.

Example

input

```
3 3
1 2 8
2 3 3
3 1 4
```

output

```
4
```

Note

After the trap is set, the new energy requirement for the first Corridor may be either smaller, larger, or equal to the old energy requiremenet.

In the example, if the energy of the first Corridor is set to 4 or less, then the Daleks may use the set of Corridors $\{\{1, 2\}, \{2, 3\}\}$ (in particular, if it were set to less than 4, then this would be the only set of Corridors that they would use). However, if it is larger than 4, then they will instead use the set $\{\{2, 3\}, \{3, 1\}\}$.

E2. Daleks' Invasion (medium)

time limit per test: 15 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

After a successful field test, Heidi is considering deploying a trap along some Corridor, possibly not the first one. She wants to avoid meeting the Daleks inside the Time Vortex, so for abundance of caution she considers placing the traps only along those Corridors that are not going to be used according to the current Daleks' plan – which is to use a minimum spanning tree of Corridors. Heidi knows that all energy requirements for different Corridors are now different, and that the Daleks have a single unique plan which they are intending to use.

Your task is to calculate the number $E_{max}(c)$, which is defined in the same way as in the easy version – i.e., the largest $e \le 10^9$ such that if we changed the energy of corridor c to e, the Daleks might use it – but now for *every* corridor that Heidi considers.

Input

The first line: number n of destinations, number m of Time Corridors ($2 \le n \le 10^5$, $n - 1 \le m \le 10^6$). The next m lines: destinations a, b and energy e ($1 \le a, b \le n$, $a \ne b$, $0 \le e \le 10^9$).

No pair $\{a, b\}$ will repeat. The graph is guaranteed to be connected. All energy requirements e are distinct.

Output

Output $m - (n - 1)$ lines, each containing one integer: $E_{max}(c_i)$ for the i-th Corridor c_i from the input that is not part of the current Daleks' plan (minimum spanning tree).

Example

input

```
3 3
1 2 8
2 3 3
3 1 4
```

output

```
4
```

Note

If $m = n - 1$, then you need not output anything.

E3. Daleks' Invasion (hard)

time limit per test: 10 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

With your help, Heidi has prepared a plan of trap placement and defence. Yet suddenly, the Doctor popped out of the TARDIS and told her that he had spied on the Daleks' preparations, and there is more of them than ever. Desperate times require desperate measures, so Heidi is going to risk meeting with the Daleks and she will consider placing a trap along any Corridor.

This means she needs your help again in calculating $E_{max}(c)$ – the largest $e \le 10^9$ such that if we changed the energy requirement of c to e, then the Daleks might use c in their invasion – but this time for all Time Corridors.

Input

First line: number n of destinations, number m of corridors ($2 \leq n \leq 10^5$, $n - 1 \leq m \leq 10^6$). The next m lines: destinations a, b and energy e ($1 \leq a, b \leq n$, $a \neq b$, $0 \leq e \leq 10^9$).

No pair $\{a, b\}$ will repeat. The graph is guaranteed to be connected. It is not guaranteed that all energy requirements e are distinct, or that the minimum spanning tree is unique.

Output

Output m lines, each containing one integer: $E_{max}(c_i)$ for the i-th Corridor c_i from the input.

Example

input

```
3 3
1 2 8
2 3 3
3 1 4
```

output

```
4
8
8
```