Problem A. DIVIDERS

Input file name:	standard input
Output file name:	standard output
Time limit:	400 ms
Memory restraint:	256 megabyte

The Lady with the Mister sat together to do a homework in mathematics. They needed to find the number of dividers of each number from **1** to **N.**

The Lady, of course, took the opportunity to diversify slightly their classes. She asked which of the number from **1** to **N** has the largest amount of dividers. The Mister should answer this question, please, help him!

Input data:

The only row in the output data is one integer **N** $(1 \le N \le 10^5)$**.**

Output data:

Come up with one row, which should contain two integers: number from **1** to **N**, that has the largest number of dividers and its amount of dividers. If there are several answers, output the smaller number.

Input file example:

21

Output file example:

12 6

Task B. ROADS

Input file name:	standard input
Output file name:	standard output
Time limit:	400 ms
Memory restraint:	256 megabytes

In the country in which resides Lady there are N towns and M roads that connect them. Lady has realized that some of the roads are unnecessary as it is enough to pick a minimum number of roads for every town to be connected and just by one route. Lady decided to find out such a road configuration.

Some of the roads are made out of paving stones and some are made out of concrete. Lady thinks that going over paving stones roads is quite interesting. But too many roads out of paved stones will impede the movement speed of the cars. Due to this, Lady wants there to be K roads made out of paving stones.

Taking into account the description of all the roads in Ukraine, help Lady find out such a configuration of roads if it exists.

Input data:

The first row contains three integers divided by one spacing: N – number of towns (1≤N≤20 000), M – number of roads (1≤M≤100 000), К – number of paved roads Lady would like to leave in the configuration (0≤К≤N-1).

The following M rows describe the roads in the country that are numbered from 1 to M. (i+1) thread describes the road with number **i**. Each of these threads has three numbers divided by a spacing:

X_i Y_i C_i

X_i and Y_i – towns connected by two-way road.

$C_i = 0$, if the road is made out of paving stones, й $C_i = 1$, if the road is made out of concrete.

It is granted that there is no more than one road connecting a single pair of towns.

Output data:

If a configuration desired by Lady does not exist you have to enter a single thread: **«no solution»** without quotation marks.

In other case your programme has to produce the number of roads included in such a configuration listing all the roads in a separate thread. The output has to contain X_i Y_i C_i of every road.

If there is more than one configuration you can choose to output **any** of them. All the roads can be output in **any order.**

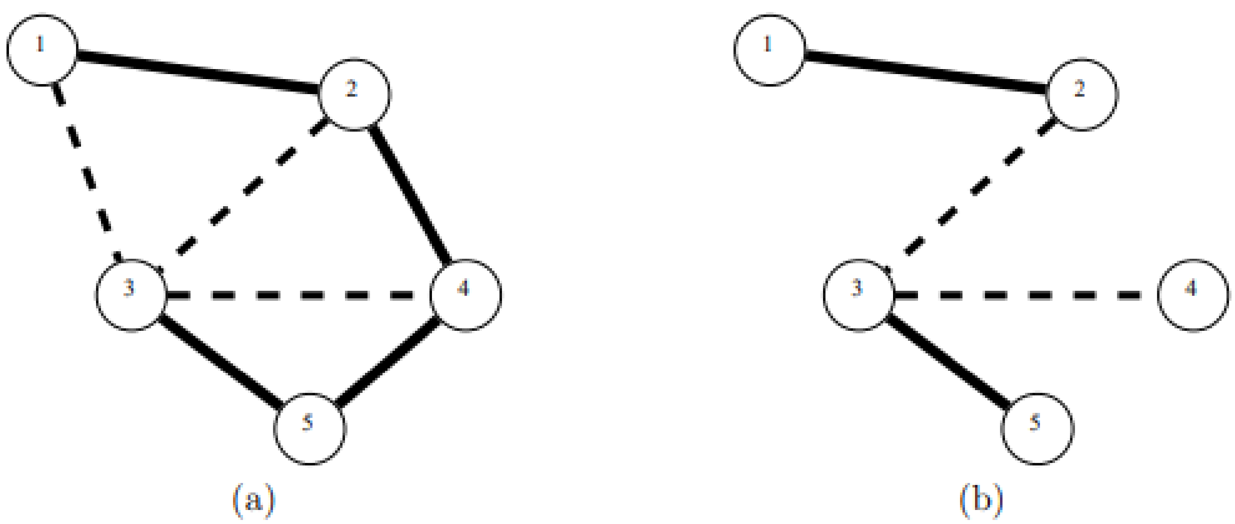
Input data example:

```
5 7 2
1 3 0
4 2 1
3 2 0
4 3 0
5 3 1
1 2 1
4 5 1
```

Output data example:

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

Explanation:



The figure (a) shows an input example. The figure (b) has only the necessary roads left in the configuration.

Problem C. GAME

Input file name:	standard input
Output file name:	standard output
Time limit:	400 ms
Memory restraint:	256 megabyte

The Lady and the Mister have two large matrixes of N * N size. The Lady fills her matrix with natural numbers in order from 1 to N * N (first, filling the first row from the left to the right, then the second one, and so on.). The Mister fills the same matrix with natural numbers in order from 1 to N * N (first, filling the first column from the top to the bottom, then the second one, and so on.).

1	2	3
4	5	6
7	8	9

1	4	7
2	5	8
3	6	9

The top figure depicts the Lady's matrix, from below the Mister's matrix for N = 3.
It turned out that some numbers, both Mister and Lady, have written in the same cell. For example, in the figure above, numbers 1, 5 and 9 coincide in both matrixes. Your task is to find all such numbers, which positions coincide in the two matrixes.

Input data

The only line of input data contains one integer N ($1 \le N \le 100$).

Output data

Deduce one row, which contains all numbers whose positions will match both matrixes. The numbers have to be displayed in ascending order, separated by a single space.

An example of input data:

3

An example of output data:

1 5 9

Problem D. Palindromes

Input file name:	standard input
Output file name:	standard output
Time limit:	2.5 second
Memory restraint:	256 megabyte

The Lady is studying programming; today she has studied a new and very interesting topic - strings. Of course, The Lady's favorite string is a palindrome. The Palindrome is a string which is read the same from both sides. For example, "aba", "gbbg", "adpda" are palindromes, and abb, gggt, ababb are not.

The lady has got a string *s* from her programming teacher. As a talented mathematician, she began to count an interesting value - the value of the string. The value of the string is the number of entries of this string into the row *s* multiplied by the length of the string. Let *s* = "abacaba". Then the value of the string "aba" = 2 (number of entries) * 3 (length) = 6 (value). The value of "c" is 1, and the "a" is 4.

The Lady tries to find the string with the highest value, but she also wants this row to be a palindrome. Find the greatest value of any palindrome for her.

Input data:

The only input row contains the *s* strip. The length of the string is marked as | s | ($1 \leq | s | \leq 300000$). It is assured that the string contains only lowercase letters of the English alphabet.

Output data:

It is necessary to deduce one number, which means the maximum value of a palindrome.

An example of input data:

abacaba

An example of output data:

7

An example of input data 2:

aaaa

An example of output data 2:

6

Problem E. DNA

Input file name:	standard input
Output file name:	standard output
Time limit:	400 ms
Memory restraint:	256 megabyte

The Lady has learned a new and interesting topic DNA at a Biology class. Everybody knows that a DNA chain is a chain of nucleotides which are represented by the characters A, C, G, T. Therefore, a DNA region can be represented by a thread of these four characters.

The Lady knew that not always can biologists identify every character in a DNA. In such a case *N* is recorded instead of one of four characters, i.e. any nucleotide can stand instead of *N*.

If there is not any character *N* in a DNA region such a succession of the DNA is called COMPLETE. Otherwise, it is INCOMPLETE. The Lady also learned from her Biology teacher that COMPLETE succession agrees with the INCOMPLETE one, if one of the main nucleotides can be put instead of every character *N*, so that these two threads become identical. For example, ACTGT agrees with ACNNT, but AGTTT does not.

The Lady habitually decided to experiment with the DNA regions. The Lady sorts four nucleotides alphabetically: A, C, G, T. The Lady also decided to classify the successions. The Lady classifies the DNA succession as form-1, if all the nucleotides are sorted in order, i.e. AACCGTTT is form-1, but AACGTC is not.

In general, the succession is a form-k for k>1, if it is a form-(k-1), or it is a concatenation of the form-(k-1) and the form-1, i.e. AACCC, ACACA, ACACC are the forms-3, but GCACAC and ACACACA are not.

The Lady likes to sort all the DNA successions lexicographically, and she is doing it right now. For example, the first succession of the form-3 is with the length 5 is AAAAA, but the last one is TTTTT.

For a different example let's consider an incomplete succession ACANNCNNG. The first seven successions of the forms-3 that agree with it are:

ACAAACAAG
ACAAACACG
ACAAACAGG
ACAAACCAG
ACAAACCCG
ACAAACCGG
ACAAACCTG

Write a programme, which will find R the form-K succession in a lexicographical order that agrees with the incomplete succession of the length M.

Input data:

The first row has 3 integer numbers divided by one space: M – length of the succession ($1 \le M \le 50\,000$), K – the form is ($1 \le K \le 10$), R is a number of the succession to find ($1 \le R \le 2 \cdot 10^{12}$).

The second row has an incomplete succession of the length M.

It is guaranteed that the number of forms-K successions that agree with the given incomplete succession is not more than $4 \cdot 10^{18}$. Moreover, R does not exceed the number of the forms-K successions which agree with the given incomplete succession.

Output data:

Come up with one thread which must contain R in the lexicographical order form-K succession which agrees with the incomplete succession M.

Example of input data:

9 3 6
ACANNCNNG

Example of output data:

ACAAACCGG

Explanation:

The statement of the problem gives first seven successions to this example.

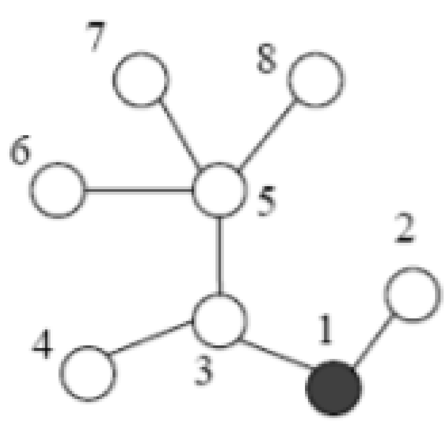
Problem F. TRIP

Input file name:	standard input
Output file name:	standard output
Time limit:	400 ms
Memory restraint:	256 megabyte

The Lady received from the Mister as a gift a graph. This graph turned out to be a tree, which is a graph with N vertices, N-1 edges, where from each vertex of the edge you can reach each other. This is a coherent graph, between each two vertices there is exactly one way. The length of each edge is equal to **1**.

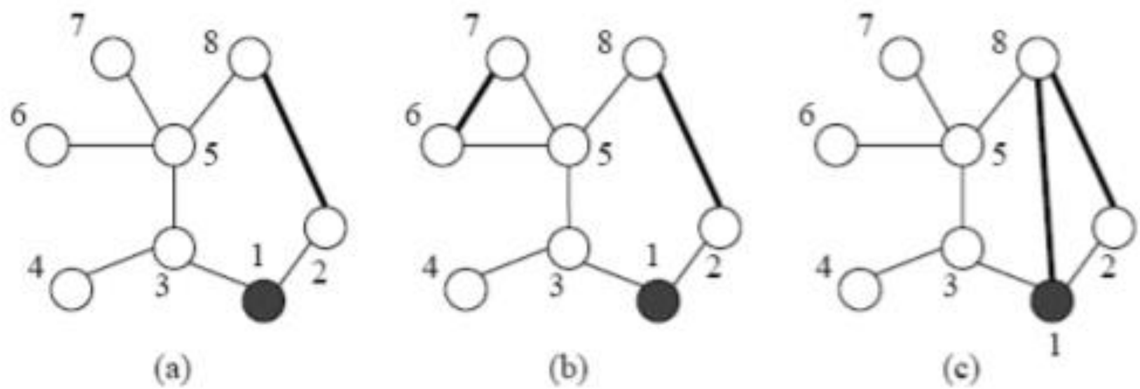
The Lady wants somehow to bypass this graph. She starts to traverse from the vertex number 1 and at the end, she returns to the vertex number 1, bypassing all the vertices one at a time. On the pass by the edge, the Lady spends no more than 1 minute.

Let us to consider an example of an eight edge's graph, as shown below. Each vertex is marked by a circle; edges are marked with the lines between them. Vertex number 1 is marked with a black circle, as the starter. To get around this graph the Lady should spend 14 minutes (that is to do 14 steps).



To reduce trip time, the Lady decided to add **K** number of new edges between some vertices. Each edge should connect any two vertices. Two edges may have one common vertex (as shown in the example (c) below).

The Lady has enough time to overcome many edges, consequently К = 1, or К = 2. The Lady decided to add another requirement – for each of the newly build roads she should pass exactly 1 time (by the other – any other). Here is an example:



On the figure (a), is added one new edge (8, 2), total trip time – 11:

1 -> 2 -> 8 -> 5 -> 7 -> 5 -> 6 -> 5 -> 3 -> 4 -> 3 -> 1

On the figure (b), is added two new edges, total trip time – 10.

1 -> 2 -> 8 -> 5 -> 7 -> 6 -> 5 -> 3 -> 4 -> 3 -> 1

On the figure (c), is added two new edges, but the total trip time is as much 15 (since it is necessary to bypass each added edge at least once).

Write the program, which according to the given graph and the number of added edges will find which edges are optimally added to the graph, so that the trip time is minimal. Find this time.

Input data:

The first row contains two integers separated by a space: N – vertex amount та K – an amount of added roads ($3 \le N \le 10^5$, $1 \le K \le 2$).

The next N-1 rows contain two numbers A та B, which means that between vertices A and B there is an edge.

Output data:

You have to come up with one number, which means the minimum trip time after adding К edges.

Input data example 1:

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

Output data example 1:

11

Input data example 2:

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

Output data example 2:

10

Input data example 3:

5 2
4 5
3 4
2 3
1 2

Output data example 3:

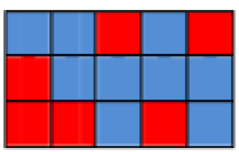
6

Task G. Coloring

Input file name:	standard input
Output file name:	standard output
Time limit:	400 ms
Memory restriction:	256 megabyte

The Lady and the Mister have a table consisting of **n×m** cells. They like to draw and paint so they decided to color in this table.

The Lady likes red and the Mister likes blue. They have been arguing how to color the table and came to the agreement it would be fair if each square of 2x2 cells had an odd number of every color (1:3). Each square of 2x2 cells may contain 1 red and 3 blue cells or vice versa. If they color the table in the specified way it is considered correct. On the figure below we can see an example of a correct matrix.



Unfortunately Mister's younger sister has painted some of the cells at night. Therefore, some of the cells may be red or blue. Now the Mister and the Lady are curious how many ways there are to paint the other cells so the rule they agreed to is maintained for every square of the table. If the sister has painted the cells in such a way that the correct matrix can not be achieved your output has to be 0.

Input data:

First line contains three numbers divided by a spacing**:** **n** – number of the row in the table ($2 \le n \le 10^5$), **m** – number of the columns in the table($2 \le m \le 10^5$), **k** – number of the cells painted by Mister's sister ($0 \le k \le 10^5$). The following **k** lines contain description of the painted cells. i+1 line contains three numbers x_i, y_i, c_i, where x_i and y_i are numbers of the row and the column of the cell painted ($1 \le x_i \le n$, $1 \le y_i \le m$), $c_i = 1$ if the cell is painted red and $c_i=0$ if the cell is painted blue. It is granted that all **k** cells are placed in different positions.

Output data:

You have to output a single line that has to contain the number of ways possible to color in the table for it to be correct. Deduce the modular answer of the 10 in 9^{th} power (10^9) (the remainder of the answer division by 10^9).

Input data example:

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

Output data example:

8

Problem H. ROAD

Input file name:	standard input
Output file name:	standard output
Time limit:	1.5 second
Memory restraint:	256 megabyte

The Mister and the Lady have found an interesting board game and began to study its rules. There is a large field in the game (we will consider it to be endless). The field of the game is the Cartesian coordinate system. There are also large buildings on the field. Each building is a rectangle and its sides are parallel to the axes of the plane of the field.

One player is involved in the game. The player can move in only four directions. If the player is in position (x, y), he can move into one of four cells in one step: (x-1, y), (x + 1, y), (x, y-1), (x, y + 1). The player can only be in the positions with integer coordinates.

There are several more rules:

- The player is not allowed to enter any building, but you can be at its edge or in the corner.

- The player can only change the direction of his movement if he is at its edge or in the corner of a building, otherwise he must continue to move in the same direction as in the last step.

- - The player begins to move in position (x_s, y_s).
- - The player must complete his movement in the position (x_f, y_f).
- - The player can start his movement in any of four directions.

The main goal of the player is to reach the endpoint as soon as possible. One step is made in one second. Changing your direction does not take time. The Lady wonders what is the minimum time to get from the starting point to the final one. Also, there are several options for the game board and the player's position. You need to find the answer for each test (option).

Input data:

The first row contains a single number - the number of the tests T ($1 \le T \le 20$). All other rows describe the T-options of the game field. Before each test there is a free row.

Each test is described as follows:

The first row contains four numbers: x_s, y_s (player's starting position), x_f, y_f (player's final position).

In the second row, the number N is the number of buildings ($0 \le N \le 1000$). The following N rows contain a description of each building. The building is described by four numbers: the coordinates of two opposite corners of the building.

It is assured that no two buildings cross, not touch either sides or corners. It is also guaranteed that the initial and final positions do not coincide. The area of each building is more than 0.

All coordinates are integer numbers which absolute value does not exceed 10^9.

Output data

Для кожного варіанту гри виведіть одне число - кількість секунд, що потрібні гравцю. **Якщо гравець не може дібратись, виведіть “No Path” без лапок.**

For each game option, deduce one number - the number of seconds the player needs.

If the player cannot reach, deduce "No Path" without quotes.

An example of input data:

2

2 1 5 4
1
3 1 4 3

1 7 7 8
2

4 10 6 7
2 5 3 8

An example of output data:

No Path
9