COCI 2018/2019

Round #3, December 15th, 2018

Tasks

Task	Time limit	Memory limit	Score
Magnus	1 s	64 MB	50
Pismo	1 s	64 MB	70
Sajam	5 s	64 MB	90
NLO	3 s	64 MB	110
Praktični	1 s	64 MB	130
Total			450

Magnus lost a game of chess to Kile so he found comfort in competitive programming. Very soon, he heard of the iconic COCI competition and decided to try his luck there.

He wrote a mail to Kile: "*Dear Kile, please, prepare me for COCI. Magnus*".
Kile replied: "*You want to participate in COCI? All right, here's your warm-up task. A series of four consecutive letters of some word that make up the subword "HONI" (Croatian acronym for COCI) is called the HONI-block. I will send you the word of length N and you will throw out as many letters as you want (it might be none as well) so that in the end there are as many HONI-blocks as possible in the word. Kile*".

Magnus was very worried and asked you, COCI competitive scene, for help. Help him determine the maximum number of HONI-blocks he can get in the final word.

INPUT

The first line contains a word of length N ($1 \le N \le 100\ 000$), consisting of uppercase letters of the English alphabet.

OUTPUT

In the first and only line print out the required number of HONI-blocks.

SAMPLE TESTS

input	input	input
MAGNUS	HHHHOOOONNNNIIII	PROHODNIHODNIK
output	**output**	**output**
0	1	2

Explanation of the second sample test:
By throwing out three letters 'H', 'O', 'N' or 'I' Magnus can get the word "HONI", which contains one HONI-block.

In a small village besides Đakovo lives Kasap. While agriculture is his branch, love and destiny, in his free time Kasap solves tasks in competitive programming and is doing very well. Particularly interesting are the tasks involving data structures.

On a sunny summer day, Kasap's friend Mirko sent him a letter we carry forward entirely:

"My dear Kasap,
I hope you tolerate well these hot summer days. I'm writing this letter because I have a problem.
One friend gave me a hard task the other day that I have not managed to solve yet. Since I know that you love this sort of tasks, I would ask you for help because I do not want to embarass myself in front of my friend. In the task there is an array A consisting of N integers. You should find an interval of the minimum value. The value of the interval [L, R] is defined as the difference between the maximum and minimum value of the numbers in that interval: max(A[L], A[L+1], …, A[R]) - min(A[L], A[L+1], …, A[R]). I will remind you that we observe only the intervals in which L is strictly less than R.
Thank you,
Mirko"

After a week of solving, Kasap has not yet managed to solve the task and asks you to help him.

INPUT

In the first line of input there is a positive integer N ($2 \le N \le 100\,000$).
In the second line of input there are N integers, which absolute value is less than 10^9.

OUTPUT

Print the minimum value of an interval.

SCORING

In the test samples worth 20 points it will hold $N \le 100$.
In the test samples worth 40 points it will hold $N \le 2\,000$.

SAMPLE TESTS

input	input	input
2 1 3	3 1 1 1	5 1 2 1 2 1
output	**output**	**output**
2	0	1

Explanation of the third test sample:
The maximum at interval [1, 5] is 2, while the minimum on the same interval is 1, so the value of that interval is 2 - 1 = 1, which is also the minimum possible value of an interval.

In the spirit of advent Milo organized his own Christmas fair. It will be the best one in Europe! The evening ends and the moment has come to turn off the lights. Some were so insolent that they didn't even deign to turn off lights on their stands! Since the electricity is more and more expensive, Milo wants all the lights to be turned off quickly. For this he will use the **l**egendary-**e**lectric-**e**lectronic-**t**ablet (**LEET**), but he also needs your help.

Milo's Christmas fair consists of stalls that are arranged in *N* rows of which every consists of *N* stands. On his **LEET** Milo has 2 buttons:

- By pressing the first button, Milo imagines one row, *x*.
  LEET then lights every stand of the x^{th} row that had been turned off, but also turns off every stand of the x^{th} row that had been turned on.
- By pressing the second button, Milo imagines one column, *x*.
  LEET then lights every stand of the x^{th} column that had been turned off, but also turns off every stand of the x^{th} column that had been turned on.

By pressing his own belly button (the "third button"), Milo will decide to walk to a particular stand and physically turn it on (or turn it off). The problem is that he has injured his leg so in order not to get a pulmonary embolism, the doctor has prescribed that the "third button" should be pressed at most *K* times ($K \le N$). Fortunately, the first and second button he can press as much as he wants.

Is it possible that Milo will shut down all the stands with arbitrary sequence of actions?

INPUT

In the first line of input there are two integer numbers *N* and *K* from the task description
($1 \le N \le 1\,000$, $0 \le K \le N$).
In the next *N* lines there are *N* characters 'x' or 'o', the initial state of the stands of the Christmas fair. The symbol 'x' represents a stand that is turned off and 'o' a stand that is turned on.

OUTPUT

Print the answer to the question from the task: "DA" (Croatian for yes) if possible or "NE" (Croatian for no) if it is not (without the quotation marks).

SCORING

In the test samples worth at least 15 points it will hold $K = 0$.
In the additional test samples worth at least 15 points N will be less or equal to 100.
In the additional test samples worth at least 30 points *K* will strictly be less than ½ *N.*

SAMPLE TESTS

input	input	input
2 0 ox ox	3 1 ooo xoo oox	4 2 oxxo xxox oxoo oxxo
output	**output**	**output**
DA	NE	DA

Explanation of the third test sample:
One possible sequence of button pressures is given after which all the stands are turned off:

- Second button (imagine column 1).
- Third button (turn on field (2, 2)).
- First button (imagine row 2).
- Second button (imagine column 4).
- Third button (turn off field (3, 3)).

The locals of the village Žabnik have been struggling for many years with unidentified flying objects (UFOs) that create circles in grain fields. The damage is particularly noticeable during summer hay mowing.

Let us imagine a rectangular grain field of N rows and M columns - the upper left field is designated by coordinates (1, 1), while the lower right field is designated by coordinates (N, M). In each field there is a certain amount of grass. Initially, amount of grass in all the fields is equal to 1. In K days UFOs of circular shape land on the field and make circles in it. On the i^{th} morning, the UFO of radius R_i with the center in the field designated by the coordinates (X_i, Y_i) lands on the field and "mows" all the grass growing on covered fields. In other words, the amount of grass in the field designated by coordinates (x, y) is reduced to 0 if it holds $(X_i - x)^2 + (Y_i - y)^2 \le R_i^2$. Each new day, with the increase of the grass, the amount of grass in all the fields increases by 1.

On K^{th} day in the evening, the locals will mow all the grass of the grain field that will be stored for feeding cattle. How much is the total amount of grass they will store?

INPUT

The first line contains positive integers N and M ($1 \le N, M \le 100\ 000$), dimensions of the grain field.

The second line contains positive integer K ($1 \le K \le 100$), the number of the days in which unidentified flying objects land to the grain field before mowing.

In the i^{th} of the following K lines there are three positive integers X_i ($1 < X_i < N$), Y_i ($1 < Y_i < M$), and R_i ($1 \le R_i \le \min(X_i - 1, Y_i - 1, N - X_i, M - Y_i)$) which represent the central field on which the i^{th} UFO lands and the radius the i^{th} UFO.

OUTPUT

Print the total amount of grass that the locals will store after mowing.

SCORING

In the sample tests worth 20% of the total points it will hold $N, M \le 1\ 000$.

SAMPLE TESTS

input
```
6 6
3
4 4 2
3 3 2
2 4 1
```
output
```
68
```

input
```
100 100
2
50 50 49
30 30 29
```
output
```
9534
```

input
```
33333 44444
1
11111 22222 9999
```
output
```
1167355751
```

Explanation of the first test sample:

The following matrix shows amount of grass in the grain field at the end of the first day:

1	1	1	1	1	1
1	1	1	**0**	1	1
1	1	**0**	**0**	**0**	1
1	**0**	**0**	**0**	**0**	**0**
1	1	**0**	**0**	**0**	1
1	1	1	**0**	1	1

The following matrix shows amount of grass in the grain field at the end of the second day:

2	2	**0**	2	2	2
2	**0**	**0**	**0**	2	2
0	**0**	**0**	**0**	**0**	2
2	**0**	**0**	**0**	1	1
2	2	**0**	1	1	2
2	2	2	1	2	2

The following matrix shows amount of grass in the grain field at the end of the third day:

3	3	1	**0**	3	3
3	1	**0**	**0**	**0**	3
1	1	1	**0**	1	3
3	1	1	1	2	2
3	3	1	2	2	3
3	3	3	2	3	3

The total amount of grass in the grain field at the end of the third day is equal to 68.

While writing an exam, Ivan had problems with the following task:
"In the input there is an integer number N. Find the N^{th} digit of the number 0.135791113151719 ..."

In order to succeed in the next attempt to pass the exam and so saving himself from repeating the academic year, he decided to practice by being the main character in tasks such as the following:

An undirected graph of *N* nodes and *M* edges is given. Each edge has a value, an integer number less than 10^9.
A simple cycle (a cycle without repeating nodes) is *good* if the bitwise XOR of all the values of the cycle's edges is equal to zero.
Ivan can make a number of *operations* on the graph. An operation consists of the following steps:

- Ivan selects an integer number *x*;
- then he selects a non-empty subset of edges of the given graph;
- and then he applies bitwise XOR by number *x* on all the the selected edges (If one of the selected edges has a value *p*, after the described operation, the new value of that edge will be equal to *p* XOR *x*)

Ivan wants to obtain a graph in which every simple cycle is good. Also, he wants to do so in as few operations as possible. Determine the minimum possible number of operations after which each simple cycle is good and print them. It can be proved that it is always possible to meet Ivan's requirements with a certain sequence of operations.

INPUT

The first line contains two positive integers *N* and *M* ($1 \le N, M \le 100\ 000$), the number of nodes and the number of edges.
In the i^{th} of the following *M* lines there is a description of the i^{th} edge: three integer numbers a_i, b_i i p_i ($1 \le a_i, b_i \le N, a_i \ne b_i, 0 \le p_i \le 10^9$), the nodes connected with the i^{th} edge and the value of the edge.
The graph is connected and all the edges are different.

OUTPUT

In the first line of output, print *K*, minimum number of task operations.
In each of the following *K* lines, print a sequence numbers separated by space:

- the first number in the row is the number *x* from the operation;
- the second number is *B*, the number of selected bridges;
- then follows *B* numbers, E_i ($1 \le E_i \le M$) which indicate labels of the selected edges in the ascending order.

All numbers in the output should be less than or equal to $2 \cdot 10^9$.

SCORING

In the test samples worth 20% of the total points, the minimum number of required operations will be equal to 1.
In the test samples worth additional 40% of the total points all the numbers from input will be less than or equal to 10^6.

SAMPLE TESTS

input

```
4 4
1 2 10
2 3 9
3 4 8
4 1 7
```

output

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

input

```
2 1
1 2 3
```

output

```
0
```

input

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

output

```
2
2 2 4 6
15 1 7
```

Explanation of test samples:

In the first test sample, the initial graph is given in the image left below, and the final graph (after applying XOR on the first three edges with 12) is given in the image right below. The only cycle in the graph is good because XOR of its edges is 0.

In the second test sample, there is no cycle, so the claim "every simple cycle is good" is trivially fulfilled. That is why the number of required operations is 0.