

## Codeforces Round #310 (Div. 2)

### A. Case of the Zeros and Ones

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Andrewid the Android is a galaxy-famous detective. In his free time he likes to think about strings containing zeros and ones.

Once he thought about a string of length  $n$  consisting of zeroes and ones. Consider the following operation: we choose any two *adjacent* positions in the string, and if one them contains 0, and the other contains 1, then we are allowed to remove these two digits from the string, obtaining a string of length  $n - 2$  as a result.

Now Andreid thinks about what is the minimum length of the string that can remain after applying the described operation several times (possibly, zero)? Help him to calculate this number.

#### Input

First line of the input contains a single integer  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ), the length of the string that Andreid has.

The second line contains the string of length  $n$  consisting only from zeros and ones.

#### Output

Output the minimum length of the string that may remain after applying the described operations several times.

#### Sample test(s)

input
4 1100
output
0
input
5 01010
output
1
input
8 11101111
output
6

#### Note

In the first sample test it is possible to change the string like the following:  $1100 \rightarrow 10 \rightarrow (\text{empty})$ .

In the second sample test it is possible to change the string like the following:  $01010 \rightarrow 010 \rightarrow 0$ .

In the third sample test it is possible to change the string like the following:  $11101111 \rightarrow 111111$ .

## B. Case of Fake Numbers

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Andrewid the Android is a galaxy-famous detective. He is now investigating a case of frauds who make fake copies of the famous Stolp's gears, puzzles that are as famous as the Rubik's cube once was.

Its most important components are a button and a line of  $n$  similar gears. Each gear has  $n$  teeth containing all numbers from  $0$  to  $n - 1$  in the counter-clockwise order. When you push a button, the first gear rotates *clockwise*, then the second gear rotates *counter-clockwise*, the the third gear rotates *clockwise* an so on.

Besides, each gear has exactly one active tooth. When a gear turns, a new active tooth is the one following after the current active tooth according to the direction of the rotation. For example, if  $n = 5$ , and the active tooth is the one containing number  $0$ , then clockwise rotation makes the tooth with number  $1$  active, or the counter-clockwise rotating makes the tooth number  $4$  active.

Andrewid remembers that the real puzzle has the following property: you can push the button multiple times in such a way that in the end the numbers on the active teeth of the gears from first to last form sequence  $0, 1, 2, \dots, n - 1$ . Write a program that determines whether the given puzzle is real or fake.

### Input

The first line contains integer  $n$  ( $1 \leq n \leq 1000$ ) — the number of gears.

The second line contains  $n$  digits  $a_1, a_2, \dots, a_n$  ( $0 \leq a_i \leq n - 1$ ) — the sequence of active teeth: the active tooth of the  $i$ -th gear contains number  $a_i$ .

### Output

In a single line print "Yes" (without the quotes), if the given Stolp's gears puzzle is real, and "No" (without the quotes) otherwise.

#### Sample test(s)

input
3 1 0 0
output
Yes
input
5 4 2 1 4 3
output
Yes
input
4 0 2 3 1
output
No

### Note

In the first sample test when you push the button for the first time, the sequence of active teeth will be  $2\ 2\ 1$ , when you push it for the second time, you get  $0\ 1\ 2$ .

## C. Case of Matryoshkas

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

Andrewid the Android is a galaxy-famous detective. He is now investigating the case of vandalism at the exhibition of contemporary art.

The main exhibit is a construction of  $n$  matryoshka dolls that can be nested one into another. The matryoshka dolls are numbered from  $1$  to  $n$ . A matryoshka with a smaller number can be nested in a matryoshka with a higher number, two matryoshkas can not be directly nested in the same doll, but there may be chain nestings, for example,  $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$ .

In one second, you can perform one of the two following operations:

- Having a matryoshka  $a$  that isn't nested in any other matryoshka and a matryoshka  $b$ , such that  $b$  doesn't contain any other matryoshka and is not nested in any other matryoshka, you may put  $a$  in  $b$ ;
- Having a matryoshka  $a$  directly contained in matryoshka  $b$ , such that  $b$  is not nested in any other matryoshka, you may get  $a$  out of  $b$ .

According to the modern aesthetic norms the matryoshka dolls on display were assembled in a specific configuration, i.e. as several separate chains of nested matryoshkas, but the criminal, following the mysterious plan, took out all the dolls and assembled them into a single large chain ( $1 \rightarrow 2 \rightarrow \dots \rightarrow n$ ). In order to continue the investigation Andrewid needs to know in what minimum time it is possible to perform this action.

### Input

The first line contains integers  $n$  ( $1 \leq n \leq 10^5$ ) and  $k$  ( $1 \leq k \leq 10^5$ ) — the number of matryoshkas and matryoshka chains in the initial configuration.

The next  $k$  lines contain the descriptions of the chains: the  $i$ -th line first contains number  $m_i$  ( $1 \leq m_i \leq n$ ), and then  $m_i$  numbers  $a_{i1}, a_{i2}, \dots, a_{im_i}$  — the numbers of matryoshkas in the chain (matryoshka  $a_{i1}$  is nested into matryoshka  $a_{i2}$ , that is nested into matryoshka  $a_{i3}$ , and so on till the matryoshka  $a_{im_i}$  that isn't nested into any other matryoshka).

It is guaranteed that  $m_1 + m_2 + \dots + m_k = n$ , the numbers of matryoshkas in all the chains are distinct, in each chain the numbers of matryoshkas follow in the ascending order.

### Output

In the single line print the minimum number of seconds needed to assemble one large chain from the initial configuration.

#### Sample test(s)

input
3 2 2 1 2 1 3
output
1

input
7 3 3 1 3 7 2 2 5 2 4 6
output
10

### Note

In the first sample test there are two chains:  $1 \rightarrow 2$  and  $3$ . In one second you can nest the first chain into the second one and get  $1 \rightarrow 2 \rightarrow 3$ .

In the second sample test you need to disassemble all the three chains into individual matryoshkas in  $2 + 1 + 1 = 4$  seconds and then assemble one big chain in 6 seconds.

## D. Case of Fugitive

time limit per test: 3 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Andrewid the Android is a galaxy-famous detective. He is now chasing a criminal hiding on the planet Oxa-5, the planet almost fully covered with water.

The only dry land there is an archipelago of  $n$  narrow islands located in a row. For more comfort let's represent them as non-intersecting segments on a straight line: island  $i$  has coordinates  $[l_i, r_i]$ , besides,  $r_i < l_{i+1}$  for  $1 \leq i \leq n - 1$ .

To reach the goal, Andrewid needs to place a bridge between each pair of **adjacent** islands. A bridge of length  $a$  can be placed between the  $i$ -th and the  $(i + 1)$ -th islands, if there are such coordinates of  $x$  and  $y$ , that  $l_i \leq x \leq r_i$ ,  $l_{i+1} \leq y \leq r_{i+1}$  and  $y - x = a$ .

The detective was supplied with  $m$  bridges, each bridge can be used at most once. Help him determine whether the bridges he got are enough to connect each pair of adjacent islands.

### Input

The first line contains integers  $n$  ( $2 \leq n \leq 2 \cdot 10^5$ ) and  $m$  ( $1 \leq m \leq 2 \cdot 10^5$ ) — the number of islands and bridges.

Next  $n$  lines each contain two integers  $l_i$  and  $r_i$  ( $1 \leq l_i \leq r_i \leq 10^{18}$ ) — the coordinates of the island endpoints.

The last line contains  $m$  **integer** numbers  $a_1, a_2, \dots, a_m$  ( $1 \leq a_i \leq 10^{18}$ ) — the lengths of the bridges that Andrewid got.

### Output

If it is impossible to place a bridge between each pair of adjacent islands in the required manner, print on a single line "No" (without the quotes), otherwise print in the first line "Yes" (without the quotes), and in the second line print  $n - 1$  numbers  $b_1, b_2, \dots, b_{n-1}$ , which mean that between islands  $i$  and  $i + 1$  there must be used a bridge number  $b_i$ .

If there are multiple correct answers, print any of them. Note that in this problem it is necessary to print "Yes" and "No" in correct case.

### Sample test(s)

input
4 4 1 4 7 8 9 10 12 14 4 5 3 8
output
Yes 2 3 1

input
2 2 11 14 17 18 2 9
output
No

input
2 1 1 1 1000000000000000000 1000000000000000000 999999999999999999
output
Yes 1

### Note

In the first sample test you can, for example, place the second bridge between points 3 and 8, place the third bridge between points 7 and 10 and place the first bridge between points 10 and 14.

In the second sample test the first bridge is too short and the second bridge is too long, so the solution doesn't exist.

## E. Case of Chocolate

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

Andrewid the Android is a galaxy-known detective. Now he does not investigate any case and is eating chocolate out of boredom.

A bar of chocolate can be presented as an  $n \times n$  table, where each cell represents one piece of chocolate. The columns of the table are numbered from 1 to  $n$  from left to right and the rows are numbered from top to bottom. Let's call the anti-diagonal to be a diagonal that goes the lower left corner to the upper right corner of the table. First Andrewid eats all the pieces lying below the anti-diagonal. Then he performs the following  $q$  actions with the remaining triangular part: first, he chooses a piece on the anti-diagonal and either direction 'up' or 'left', and then he begins to eat all the pieces starting from the selected cell, moving in the selected direction until he reaches the already eaten piece or chocolate bar edge.

After each action, he wants to know how many pieces he ate as a result of this action.

### Input

The first line contains integers  $n$  ( $1 \leq n \leq 10^9$ ) and  $q$  ( $1 \leq q \leq 2 \cdot 10^5$ ) — the size of the chocolate bar and the number of actions.

Next  $q$  lines contain the descriptions of the actions: the  $i$ -th of them contains numbers  $x_i$  and  $y_i$  ( $1 \leq x_i, y_i \leq n$ ,  $x_i + y_i = n + 1$ ) — the numbers of the column and row of the chosen cell and the character that represents the direction (L — left, U — up).

### Output

Print  $q$  lines, the  $i$ -th of them should contain the number of eaten pieces as a result of the  $i$ -th action.

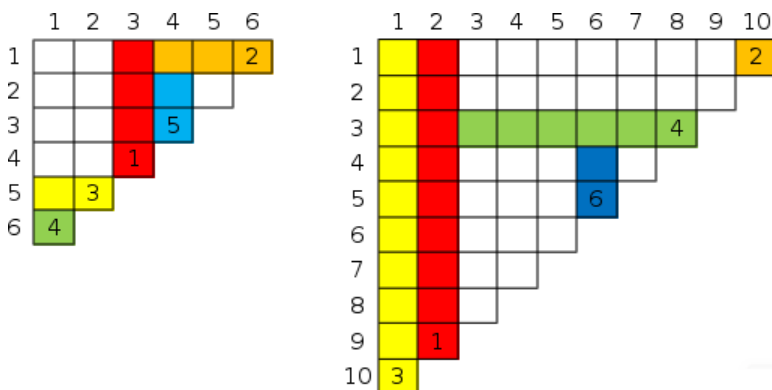
### Sample test(s)

input
6 5 3 4 U 6 1 L 2 5 L 1 6 U 4 3 U
output
4 3 2 1 2

input
10 6 2 9 U 10 1 U 1 10 U 8 3 L 10 1 L 6 5 U
output
9 1 10 6 0 2

### Note

Pictures to the sample tests:



The pieces that were eaten in the same action are painted the same color. The pieces lying on the anti-diagonal contain the numbers of the action as a result of which these pieces were eaten.

In the second sample test the Andrewid tries to start eating chocolate for the second time during his fifth action, starting from the cell at the intersection of the 10-th column and the 1-st row, but this cell is already empty, so he does not eat anything.