A. Start Up

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

Recently, a start up by two students of a state university of city F gained incredible popularity. Now it's time to start a new company. But what do we call it?

The market analysts came up with a very smart plan: the name of the company should be identical to its reflection in a mirror! In other words, if we write out the name of the company on a piece of paper in a line (horizontally, from left to right) with large English letters, then put this piece of paper in front of the mirror, then the reflection of the name in the mirror should perfectly match the line written on the piece of paper.

There are many suggestions for the company name, so coming up to the mirror with a piece of paper for each name wouldn't be sensible. The founders of the company decided to automatize this process. They asked you to write a program that can, given a word, determine whether the word is a 'mirror' word or not.

Input

The first line contains a non-empty name that needs to be checked. The name contains at most 10^5 large English letters. The name will be written with the next sans serif font:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

Output

Print 'YES' (without the quotes), if the given name matches its mirror reflection. Otherwise, print 'NO' (without the quotes).

Examples

input
AHA
output
YES

input
Z
output
NO

input
XO
output
NO

B. Online Meeting

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

Nearly each project of the F company has a whole team of developers working on it. They often are in different rooms of the office in different cities and even countries. To keep in touch and track the results of the project, the F company conducts shared online meetings in a Spyke chat.

One day the director of the F company got hold of the records of a part of an online meeting of one successful team. The director watched the record and wanted to talk to the team leader. But how can he tell who the leader is? The director logically supposed that the leader is the person who is present at any conversation during a chat meeting. In other words, if at some moment of time at least one person is present on the meeting, then the leader is present on the meeting.

You are the assistant director. Given the 'user logged on'/'user logged off' messages of the meeting in the chronological order, help the director determine who can be the leader. Note that the director has the record of only a continuous part of the meeting (probably, it's not the whole meeting).

Input

The first line contains integers n and m ($1 \le n, m \le 10^5$) — the number of team participants and the number of messages. Each of the next m lines contains a message in the format:

- '+ id': the record means that the person with number id ($1 \le id \le n$) has logged on to the meeting.
- '- id': the record means that the person with number id ($1 \le id \le n$) has logged off from the meeting.

Assume that all the people of the team are numbered from 1 to n and the messages are given in the chronological order. It is guaranteed that the given sequence is the correct record of a continuous part of the meeting. It is guaranteed that no two log on/log off events occurred simultaneously.

Output

In the first line print integer k ($0 \le k \le n$) — how many people can be leaders. In the next line, print k integers in the increasing order — the numbers of the people who can be leaders.

If the data is such that no member of the team can be a leader, print a single number 0.

Examples

input
5 4 + 1 + 2 - 2 - 1
output
4 1 3 4 5

input
3 2 + 1 - 2
output
1 3

input
2 4 + 1 - 1 + 2 - 2
output
0

input
5 6 + 1 - 1 - 3

+ 3
+ 4
- 4

output

3
2 3 5

input

2 4
+ 1
- 2
+ 2
- 1

output

0

C. Bug in Code

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

Recently a serious bug has been found in the FOS code. The head of the F company wants to find the culprit and punish him. For that, he set up an organizational meeting, the issue is: who's bugged the code? Each of the n coders on the meeting said: 'I know for sure that either x or y did it!'

The head of the company decided to choose two suspects and invite them to his office. Naturally, he should consider the coders' opinions. That's why the head wants to make such a choice that at least p of n coders agreed with it. A coder agrees with the choice of two suspects if at least one of the two people that he named at the meeting was chosen as a suspect. In how many ways can the head of F choose two suspects?

Note that even if some coder was chosen as a suspect, he can agree with the head's choice if he named the other chosen coder at the meeting.

Input

The first line contains integers n and p ($3 \le n \le 3 \cdot 10^5$; $0 \le p \le n$) — the number of coders in the F company and the minimum number of agreed people.

Each of the next n lines contains two integers x_i, y_i ($1 \le x_i, y_i \le n$) — the numbers of coders named by the i-th coder. It is guaranteed that $x_i \ne i$, $y_i \ne i$, $x_i \ne y_i$.

Output

Print a single integer –– the number of possible two-suspect sets. Note that the order of the suspects doesn't matter, that is, sets $(1, 2)$ и $(2, 1)$ are considered identical.

Examples

input
4 2 2 3 1 4 1 4 2 1
output
6

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

D. Cup Trick

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

The employees of the F company have lots of ways to entertain themselves. Today they invited a famous magician who shows a trick with plastic cups and a marble.

The point is to trick the spectator's attention. Initially, the spectator stands in front of a line of n plastic cups. Then the magician places a small marble under one cup and shuffles the cups. Then the spectator should guess which cup hides the marble.

But the head coder of the F company isn't easy to trick. When he saw the performance, he noticed several important facts:

- each cup contains a mark — a number from 1 to n; all marks on the cups are distinct;
- the magician shuffles the cups in m operations, each operation looks like that: take a cup marked x_i, sitting at position y_i in the row of cups (the positions are numbered from left to right, starting from 1) and shift it to the very beginning of the cup row (on the first position).

When the head coder came home after work he wanted to re-do the trick. Unfortunately, he didn't remember the starting or the final position of the cups. He only remembered which operations the magician performed. Help the coder: given the operations in the order they were made find at least one initial permutation of the cups that can go through the described operations in the given order. Otherwise, state that such permutation doesn't exist.

Input

The first line contains integers n and m ($1 \le n, m \le 10^6$). Each of the next m lines contains a couple of integers. The i-th line contains integers x_i, y_i ($1 \le x_i, y_i \le n$) — the description of the i-th operation of the magician. Note that the operations are given in the order in which the magician made them and the coder wants to make them in the same order.

Output

If the described permutation doesn't exist (the programmer remembered wrong operations), print -1. Otherwise, print n distinct integers, each from 1 to n: the i-th number should represent the mark on the cup that initially is in the row in position i.

If there are multiple correct answers, you should print the lexicographically minimum one.

Examples

input
2 1 2 1
output
2 1

input
3 2 1 2 1 1
output
2 1 3

input
3 3 1 3 2 3 1 3
output
-1

E. Playing the ball

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

A coder cannot sit and code all day. Sometimes it is a good idea to rise from the desk, have a rest, have small talk with colleagues and even play. The coders of the F company have their favorite ball game.

Let's imagine the game on the plane with a cartesian coordinate system. The point (0, 0) contains the player who chooses an arbitrary direction and throws a ball in that direction. The ball hits the plane at distance d from the player's original position and continues flying in the same direction. After the ball hits the plane for the first time, it flies on and hits the plane again at distance $2 \cdot d$ from the player's original position and so on (it continue flying in the chosen direction and hitting the plane after each d units). All coders in the F company are strong, so the ball flies infinitely far away.

The plane has n circles painted on it. If a ball hits the plane and hits a circle that is painted on the plane (including its border), then the player gets one point. The ball can hit multiple circles at once and get one point for each of them (if the ball hits some circle x times during the move, the player also gets x points). Count the maximum number of points a player can get if he throws a ball in the arbitrary direction. Note that the direction may have real cooridinates.

Input

The first line contains two space-separated integers — n и d ($1 \le n \le 2 \cdot 10^4$; $5 \le d \le 10$). Next n lines contain the circles' description. The i-th line contains three space-separated integers x_i, y_i, r_i (- 10000 $\le x_i, y_i \le$ 10000; $1 \le r \le 50$), where (x_i, y_i, r_i) are the coordinates of the center and the radius of the circle, correspondingly. The point (0, 0) is not inside or on the border of some circle.

Output

Print a single integer — the maximum number of points you can get.

Examples

input
2 5 1 1 1 5 0 1
output
1

input
2 5 4 0 3 5 3 1
output
2

input
1 10 20 0 10
output
3