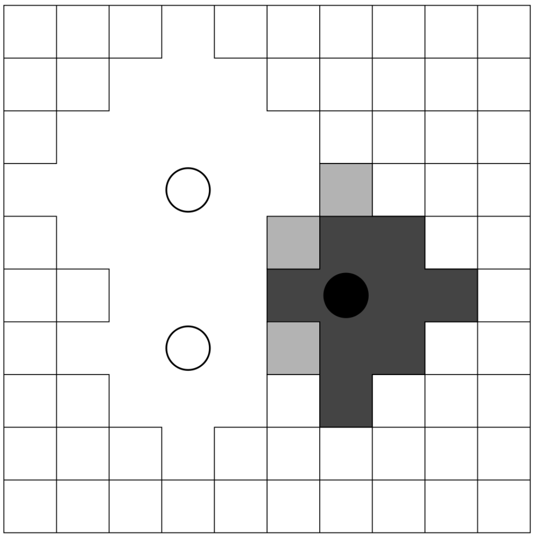
Dominance

Imagine the land of Bugtopia as a grid of $W \times H$ squares. Bugtopia is inhabited by white and black bugs. Each square is either inhabited by white bugs only (we call it "white square" then) or by black bugs only (a "black square") or not inhabited at all. White bugs are pretty aggressive against black bugs, and vice versa. Each kind wants to dominate Bugtopia. For that purpose, the bugs move along the grid; a move to a vertically or horizontally adjacent square is counted as one step. Thus, the bugs of one square are able to attack other squares if they can reach them with no more than a certain number of steps. This "range" depends on their square; different squares provide different living conditions. We say that a square is *dominated* by the white bugs if it can be attacked from more white squares than black squares. Similarly, a square is *dominated* by the black bugs if it can be attacked from more black than white squares. Note that a square is called *neutral* if it can be attacked from no square or from equally many white and black squares.



This picture shows two white squares (marked with a white circle) with ranges 3 and 2, as well as one black square (marked with a black circle) with range 2. The white bugs dominate 30 squares, the black bugs dominate 9 squares. The three squares colored in light gray can be attacked, but are neutral and therefore not dominated by any kind of bugs.

Given the size of the grid and the positions, colors, and ranges of the inhabited squares, output the total number of squares dominated by each kind of bugs.

Input

The first line contains two integers, W and H, that determine the size of the grid ($1 \leq W, H \leq 1\,000\,000\,000$). The next line contains a single integer N ($0 \leq N \leq 3\,000$), the number of inhabited squares.

The following N lines each describe an inhabited square. That is, line $i+2$ contains values c_i, x_i ($0 \leq x_i < W$), y_i ($0 \leq y_i < H$), and r_i ($0 \leq r_i < 500\,000\,000$), separated by single spaces: The square's color, its coordinates, and its range, respectively. The square's color is either 'W' (a white square) or 'B' (a black square). Note that the squares' ranges will never reach beyond the borders of Bugtopia. The bottom-left square of the grid has coordinates $(0, 0)$, and the top-right square has coordinates $(W - 1, H - 1)$.

In at least 30% of the test cases $W, H \leq 2\,000$.

Output

Output one line with two integers separated by a single space: the number of squares dominated by the white bugs, followed by the number of squares dominated by the black bugs.

Example

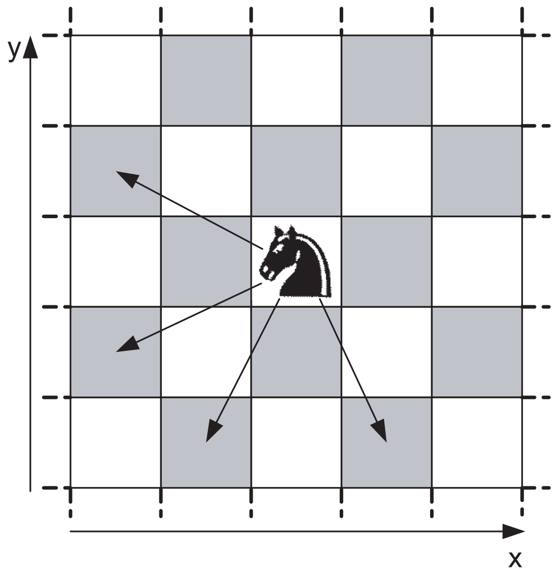
This example corresponds to the picture above.

Example

Standard input	Standard output
10 10 3 W 3 6 3 B 6 4 2 W 3 3 2	30 9

Knights

Alice and Bob are playing a game. Initially K black Knights are placed on a $N \times N$ chessboard. Now the players take turns. On each turn, a player moves every knight that has at least one valid move left. The following four moves are valid, as long as they do not move the knight off the board:



A knight with no valid moves left remains at its current position. The first player who is not able to move any of the knights loses the game. Note that during the game several knights are allowed to occupy the same square.

You are given the locations of the knights on the chessboard. Alice begins the game. Determine whether she can win the game, assuming that both players play optimally. If she can win, output a possible first move for each knight. In the beginning, there is at least one valid move for each knight, and no two knights are placed on the same square of the chessboard.

Input

The first line contains the two integers K ($1 \leq K \leq 200\,000$) and N ($1 \leq N \leq 1\,000\,000$) separated by a single space. This line is followed by K lines describing the positions of the knights.

Line $i+1$ contains two integers x_i and y_i ($1 \leq x_i,\ y_i \leq N$) separated by a single space, the coordinates of knight i.

Output

Output a single line containing the word `NO` if Alice cannot win the game. Otherwise, output $K+1$ lines with `YES` on the first line and coordinates x'_i, y'_i on line $i+1$, giving a destination that Alice may choose for knight i in the first turn in order to win the game.

Example

Standard input	Standard output
2 3 2 3 3 2	YES 1 1 1 1
3 4 2 3 3 2 4 4	NO

Information

A secret intelligence agency (which is even too secret to mention its name here) controls *agents* around the world. From time to time the headquarters need to send out a message to all agents. For obvious reasons, the transmission must be as secure as possible.

The agency's executives mistrust electronic communication and therefore transfer their messages by contact persons (in short: *contacts*). They organized agents and contacts into a large network; each contact is responsible for transporting information from exactly one agent to another, and only in this one direction. Nonetheless there might be more than one contact to transport information between two agents.

When the headquarters send out a message, their "message officer" uses some of his own contacts to transport it to a number of field agents. Those agents use their contacts to forward the message to other agents, and so on until it eventually reaches every single agent. However, in order to reduce risk, the number of times a message is transported by contacts should be minimized (i.e. no agent should receive the same message twice). Therefore an agent doesn't forward a message using all of his contacts but obeys a "transmission scheme" for this message. A *transmission scheme* contains information on how a message is to be forwarded by the agents.

Recently, the agency found out that some contacts misused confidential information. For this reason, they decided to split each message into two *parts* which are both useless if not read together. They now send out the two parts but without using the same contact twice. Therefore no contact will see the full message. Nonetheless it is important that every agent eventually receives both parts of the message. The agency now wonders how to create *valid* transmission schemes for each part that satisfy the above conditions.

Task

Write a program that calculates valid transmission schemes for each part of the message, given the agency's network of agents and contacts. It might be the case that no such two schemes exist.

Input

The first line of the input contains two integers N ($2 \leq N \leq 2\,000$), the number of agents, and M ($1 \leq M \leq 1\,000\,000$), the number of contact persons. The message officer in the headquarters has the number 1, the other agents are numbered from 2 to N; contacts are numbered from 1 to M. The i-th of the next M lines contains two integers v_i and $w_i \neq v_i$, describing the fact that contact i knows how to deliver a message from agent v_i to agent w_i.

Output

If no two valid transmission schemes exist, the output consists of a single line with the string `NONE`.

Otherwise, the output consists of two lines. Each line describes a valid transmission scheme for one part of the message by giving the list of contacts used to transmit it; the first line for the first part, the second line for the second part. If there is more than one solution, output any of them.

Example

Standard input	Standard output
4 6 1 2 1 3 2 3 3 2 2 4 2 4	1 3 5 2 4 6

Explanation

The first part is transmitted using the contacts 1, 3 and 5, i.e. from the headquarters to agent 2, from agent 2 to agent 3 and from agent 2 also to agent 4. The second part is transmitted using the contact persons 2, 4 and 6, i.e. from the headquarters to agent 3, from agent 3 to agent 2 and from agent 2 to agent 4.