CCC 2013 Stage 2
Day 2, Problem 1: A Romantic Movie Outing

Problem Description

Brian the Computer Science Nerd is going on a date with his girlfriend, Anatevka! His romantic location of choice is a movie theatre - but not an IMAX theatre, of course, as that would be far too expensive.

This theatre has 10^9 rows of 1000 seats each, which are initially empty. The rows are numbered $1..10^9$ starting from the one closest to the screen, and the seats in each row are numbered $1..1000$ from left to right. Seat c in row r is denoted as seat (r,c). Seats in rows $1..L$ ($1 \leq L \leq 1000$) are considered to be "close" to the screen, while seats in further rows are considered to be "far".

Over the course of T ($1 \leq T \leq 500\ 000$) minutes before the movie starts, a number of events occur. During the ith minute, either a person enters and sits in the empty seat (R_i,C_i), the person sitting in the occupied seat (R_i,C_i) leaves, or Anatevka suggests that she and Brian take seats (R_i,C_i) and (R_i,$C_i + 1$). The type of the ith event is represented by the character E_i, with $E_i =$ "E" indicating a person entering, $E_i =$ "L" indicating a person leaving, and $E_i =$ "S" indicating a seating suggestion. All seats involved in the events are valid seats inside the theatre, and every seat that Anatevka suggests will be "close", as she believes that they're the best.

Every time Anatevka makes a suggestion, Brian must, of course, analyze its quality. If either of the two seats she suggests is already occupied, he should explain that her recommendation is invalid with a simple "No". Otherwise, he'd like to calculate the total inconvenience of both seats in such an arrangement. The inconvenience of sitting in seat (r,c) is the number of occupied seats in its field of vision (excluding itself). The field of vision of seat (r,c) includes all seats which are no further than it from seat (1,c) by Manhattan distance (i.e., Manhattan distance between (x_1, y_1) and (x_2, y_2) is $|x_1 - x_2| + |y_1 - y_2|$), as shown below (with the "S" representing a suggested seat, and an "F" representing a seat within its field of vision):

			S			
		F	F	F		
	F	F	F	F	F	
F	F	F	F	F	F	F

After all of the events have taken place, the movie is about to start, and a final decision must be made on where to sit - and Brian will handle that. He concludes that seats that are "far" are clearly superior (as they offer a broader view of the screen), and he knows that the point of going to the movies is to have an optimal viewing experience, so selecting two adjacent seats is certainly not mandatory. As such, he'd like to determine the minimum total inconvenience for any two "far", unoccupied seats in the theatre. Note that, if one of the chosen seats is in the other's field of vision, this does not count towards its inconvenience - it's only determined by other people sitting in the theatre.

Input Specification

The first line of input contains two integers, L ($1 \leq L \leq 1000$) and T ($1 \leq T \leq 500000$).

The next T lines each contain one character, E_i ($E_i \in \{\mathrm{E}, \mathrm{L}, \mathrm{S}\}$), and two integers, R_i and C_i, for $i = 1..T$ ($1 \leq R_i \leq 1\,000\,000\,000$; $1 \leq C_i \leq 1000$).

For test cases worth 20% of the points, you may assume $L \leq 100$ and $T \leq 400$.

For test cases worth 60% of the points, you may assume $L \leq 500$ and $T \leq 50000$.

Output Specification

For each of Anatevka's suggestions (i.e., when $E_i = \mathrm{S}$ in the input), output the string `No` if the suggestion is invalid; otherwise, output the total inconvenience of the two suggested seats.

The last line of output should contain the minimum total inconvenience of any pair of "far", unoccupied seats.

Sample Input

```
3 7
E 1 2
E 2 5
S 3 4
E 2 3
L 2 5
S 1 3
S 2 2
```

Output for Sample Input

```
3
0
No
0
```

Explanation of Output for Sample Input

When Anatevka makes her first suggestion, the front 3 rows and leftmost 5 columns of the theatre look as follows (where a “P” represents a person, and an “S” represents one of the suggested seats):

			S	S
				P
	P			

The person sitting in seat (1,2) is in the field of vision of both suggested seats, while the person sitting in seat (2,5) is only in the way of the right one. As such, the total inconvenience of the two seats is $1 + 2 = 3$.

The second suggestion is shown below:

		P		
	P	S	S	

These two seats aren’t obstructed by any people, so their total inconvenience is 0. The final suggestion is invalid, as one of its two seats (seat (2,3)) is already occupied.

Finally, Brian can easily select two “far” which each have inconvenience 0, as the theatre has $10^9 - 3$ “far” rows with 1000 seats each, and most are far from the two people setting in the theatre after the last event. For example, he might choose to take seat (4,6), while recommending that Anatevka enjoy the view from seat (100,1000).

CCC 2013 Stage 2
Day 2, Problem 2: Transforming Comets

Problem Description
While traveling from Earth to Krypton, Superman was caught in a wormhole and transported instantly to some unknown location. Superman remembers seeing periodic comets from Earth, and he can see some periodic comets from his current location. He would like to use these comets to find his bearings, but first he has to match up which one is which.

The specific comets Superman sees are periodic Gaussian hyper-comets. A periodic Gaussian hyper-comet is a sequence $(p_1, p_2, \ldots, p_n)$ where each p_i is a 2-dimensional point (x_i, y_i) with integer coordinates. The comet visits some point p_i and then it visits point p_{i+1}. The sequence is periodic: after visiting p_n the comet visits p_1 next (so the indices are interpreted modulo n), Gaussian hyper-comets also have the special property that $p_i \neq p_{i+1}$ for each i and $p_1 \neq p_n$.

Superman was disoriented in both space and time. In terms of space this means a comet that he saw before might now have its whole set of points *rotated*, *both axes scaled by the same positive factor*, and/or *translated*. Furthermore, since he was disoriented in time, the first point of a comet he used to know might not be the first point any more.

For example, the right-triangular hyper-comet ((0, 0), (1, 0), (0, 1)) from earth might look like ((40, 40), (20, 20), (60, 20)) or ((20, 20), (60, 20), (40, 40)). Note that reversing time or space is not an allowed transformation, e.g. it's **not** possible for this hyper-comet to appear as ((0, 1), (1, 0), (0, 0)).

Your goal is: given a periodic sequence of points corresponding to a Gaussian hyper-comet Superman saw from earth, and a Gaussian hyper-comet Superman sees now, determine if they could be the same comet.

Input Specification
The first line contains $1 \leq t \leq 10$, the number of test cases to follow.

Each test case begins with an integer n where $2 \leq n \leq 500\,000$. The following n lines, for i from 1 to n, each contain a space-separated pair of integers $x_i \;\; y_i$. These lines denote the points for a sequence viewed from Earth. Then, similarly, there are n more lines each containing a pair of integers $x'_i \;\; y'_i$; these lines denote the points for a sequence viewed Superman's current location.

It is guaranteed that all of the coordinates are integers between 0 and $30\,000$ (inclusive).

Output Specification
For each test case, if the two sequences could represent the same hyper-comet under this disorientation process, print out the smallest positive integer s so that $x_1 \;\; y_1$ could correspond to $x'_s \;\; y'_s$. If there is no such s, print 0.

Sample Input

```
3
3
0 0
1 0
0 1
20 20
60 20
40 40
4
0 0
1 1
0 0
1 1
30 30
19 23
30 30
19 23
4
0 0
1 0
1 1
0 1
0 0
2 0
2 1
0 1
```

Sample Output

```
3
1
0
```

CCC 2013 Stage 2
Day 2, Problem 3: Repetitivity

Problem Description

Any string of length n has 2^n subsequences, which are the strings obtained by deleting some subset of the characters. But these subsequences may not all be distinct. For example, the string "zoo" has only 6 *distinct* subsequences:

- the subsequences "z", "oo", and "zoo" appear only once,
- the empty subsequence appears only once,
- and the subsequences "o" and "zo" each appear twice.

Suppose a string S has k distinct subsequences, and that the ith one appears f_i times. Then the *repetitivity* of S is defined as $\sum_{i=1}^{k} f_i^2$. For example, the repetitivity of "zoo" is

$$1^2 + 1^2 + 1^2 + 1^2 + 2^2 + 2^2 = 12.$$

Input Specification

Each test case contains a line containing the string S (with length at most 10 000), followed by a line containing a single integer M ($2 \leq M \leq 1\ 000\ 000\ 000$). You may assume that S only contains characters with ASCII codes between 33 and 126 inclusive (these are all printable, non-whitespace characters).

For test cases worth 20% of the points, you may assume that S will be at most 20 characters long.

Output Specification

The output should consist of a single line, containing the repetitivity of S, modulo M.

Sample Input 1

```
zoo
10
```

Output for Sample Input 1

```
2
```

Sample Input 2

```
@#$%
1000000
```

Output for Sample Input 2

```
16
```