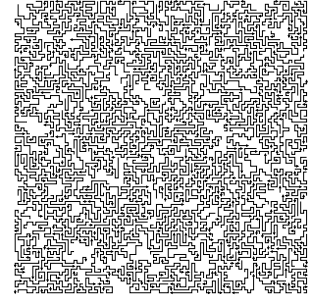


I Irritating Interruption I

Time limit: 5s

Like many people, Tonia is frequently plagued by mobile game ads. He is just watching some videos, ignoring the ads, when he notices something strange in one of them.

The game is quite simple: given some arrows in a rectangular grid, the goal is to remove all arrows. An arrow may be removed if it is not pointing at any arrow, including itself (i.e. no part of any arrow is straight in front of the arrow head at any distance). However, Tonia is fairly sure that it is not possible to remove all arrows!



One ad showed Tonia this grid.

This interruption has already taken too much time. He is now behind on his anime episodes, which means he is behind on going to bed, and thus on getting up and working on his thesis. It is, frankly, rather irritating. Write a program to help him determine how many arrows can actually be removed.

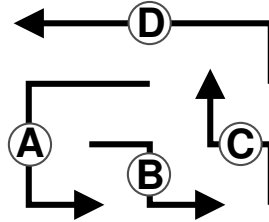


Figure I.1: Visualization of Sample Input 1. Arrows are given in the order $ABCD$ in the input. Since arrow A points at both B and C , B points at C , and C points at D , the arrows may only be removed in the order $DCBA$.

Input

The input consists of:

- One line with two integers w and h ($1 \leq w, h \leq 10^6$, $2 \leq w \cdot h \leq 10^6$), the width and height of the grid.
- One line with one integer k ($1 \leq k \leq \frac{w \cdot h}{2}$), the number of arrows.
- $2k$ lines, each pair of lines describing an arrow:
 - One line with two integers x and y ($0 \leq x < w$, $0 \leq y < h$), the coordinates of the tail of the arrow. The origin $(0,0)$ is located in the bottom left.
 - One line with a string of characters c ($c \in \{U, D, L, R\}$), indicating the direction of the arrow from the tail onwards (up/down/left/right respectively, or the directions $+y/-y/-x/+x$). It is guaranteed the arrows do not leave the grid. They also do not overlap: any coordinate may only be part of at most one arrow.

Output

Output the number of arrows that can be removed.

Sample Input 1

```
5 4
4
2 2
LLDDR
1 1
RDR
4 0
ULU
4 2
ULLLL
```

Sample Output 1

```
4
```

Sample Input 2

```
3 3
2
0 2
DDR
2 0
UUL
```

Sample Output 2

```
0
```