Grid building system

The experimental rectangular grid consists of M rows of cells, each rows contains N cells. The building system is installed above the grid and its goal is to place a number of flat wooden block on the grid according to the rules described below.
There are four principal directions on the grid -- East, North, West and South. The index of the uppermost (northernmost) row is 0 and the index of the leftmost (westernmost) column is 0.

The building system has a unlimited source of blocks. Each block is a square which side lengths are multiples of the length of one cell side in the grid. Thus, there are blocks of size 1×1, 2×2, 3×3, etc.
When a block is positioned on the grid its sides are aligned with the cell borders. The position of the block is specified by the position of its upper left corner, which is its northwest corner.

The building system recognizes two commands - Put and Move. The Put command puts a block of the specified size on the specified position on the grid. The Move command moves the specified block on the grid by one cell in the specified direction.

The format of the Move command is
M *row column direction*
The block which covers the cell at position (*row*, *column*) is moved by one cell in the specified direction. If no block covers the cell at position (*row*, *column*) the command is not executed. The parameter *direction* is one of the four symbols E, N, W, S, representing, in this order, east, north, west, south.
The format of the Put command is
P *row column size*
A block of size *size* is put on the grid so that its upper left corner is at the cell with coordinates(*row*, *column*).

If the result of a command would be a block overlapping another block the command is not executed. Also, if the result of the command would be a block positioned partially or completely outside the grid borders the command is not executed. Two blocks overlap when they share at least one cell.
For example, when the east edge of one block touches a non-zero length of the west edge of another block, it is impossible to move the first block to the east and it is impossible to move the other block to the west. When such a command is issued it is not executed.
For another example, if a block of size 3×3 is positioned at position (4,4) no block can be put on positions (5,4), (4,5), (5,5), (4,6) (5,6), (6,6), (6,5), (5,4). Also, in this case, a block of size 2×2 or bigger cannot be put at position (3,3).

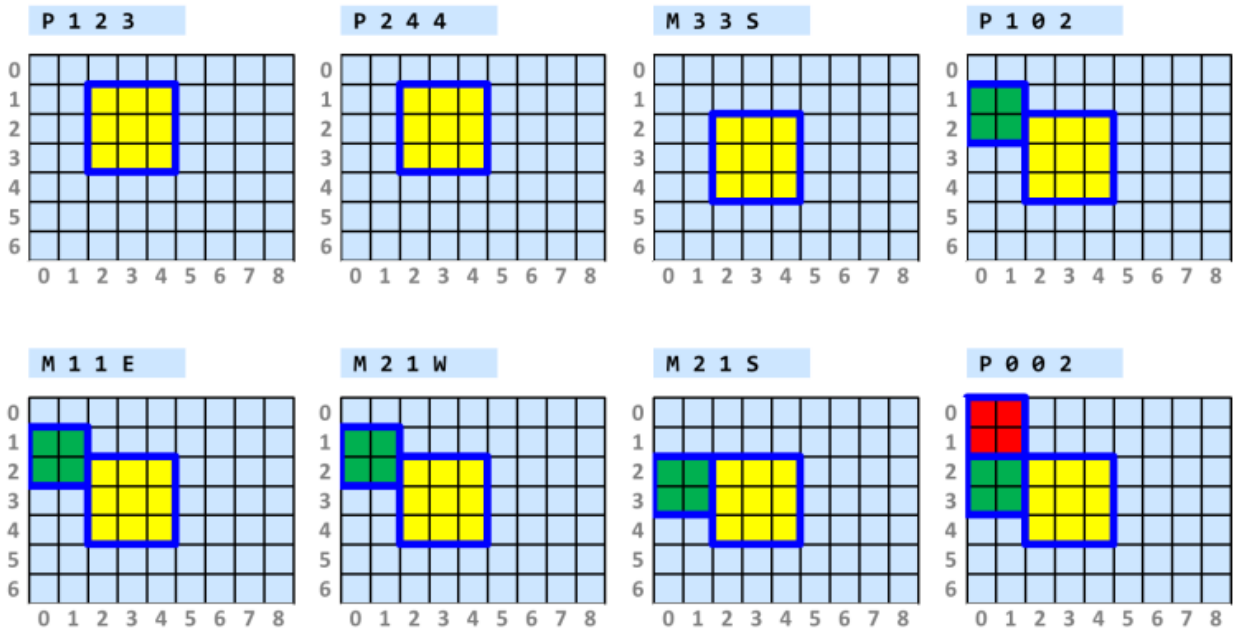

Image 1. An illustration of the effect of a sequence of 8 commands in a grid with 7 rows and 9 columns. The commands are issued in order from left to right and then from top to bottm. The grid under each command depicts the situation after the command has been issued. Note that in some cases the command has no effect, according to the rules stated in the text. The image depicts Example 1 below.

The task

You are given the size of an initially empty grid and the sequence of commands. Calculate the number of cells which are not covered by any block after all commands have been issued.

Input

The first input line contains three integers R, C and N, representing the number of rows in the grid, the number of columns in the grid, and the number of commands that follow. The rows and columns are indexed from 0 to M−1 and N−1, respectively.
The next N lines contain the list of commands, each line contains one command. The commands are in the format given above, command parameters are separated by spaces.
It holds, $2 \leq R, C \leq 200$, $1 \leq N \leq 500$. The cell coordinates specified in the commands are never outside the grid.

Output

The output consists of a single text line containing the total number of cells uncovered by any block on the grid after all input commands has been issued.

Example 1

Input

```
7 9 8
P 1 2 3
P 2 4 4
M 3 3 S
P 1 0 2
M 1 1 E
M 2 1 W
M 2 1 S
P 0 0 2
```

Output

```
46
```

Example 2

Input

```
6 6 10
P 1 1 5
P 2 2 5
M 3 3 N
M 0 5 W
P 3 5 1
M 3 5 N
M 3 5 N
P 3 5 1
M 3 5 N
P 3 5 1
```

Output

```
9
```

Example 3

Input

```
2 8 5
P 0 5 2
P 0 2 2
P 0 0 2
M 1 6 W
P 0 6 2
```

Output

```
0
```

Illustration of Example 2 and 3

```
The situation on the grid after each command in Examples 2 and 3.
Empty cells are depicted by dots, separate blocks are marked by different
numbers in order in which they are put on the grid.

Example 2
```

```
P 1 1 5          P 2 2 5          M 3 3 N          M 0 5 W          P 3 5 1
0| . . . . . .   0| . . . . . .   0| . 1 1 1 1 1   0| 1 1 1 1 1 .   0| 1 1 1 1 1 .
1| . 1 1 1 1 1   1| . 1 1 1 1 1   1| . 1 1 1 1 1   1| 1 1 1 1 1 .   1| 1 1 1 1 1 .
2| . 1 1 1 1 1   2| . 1 1 1 1 1   2| . 1 1 1 1 1   2| 1 1 1 1 1 .   2| 1 1 1 1 1 .
3| . 1 1 1 1 1   3| . 1 1 1 1 1   3| . 1 1 1 1 1   3| 1 1 1 1 1 .   3| 1 1 1 1 1 2
4| . 1 1 1 1 1   4| . 1 1 1 1 1   4| . 1 1 1 1 1   4| 1 1 1 1 1 .   4| 1 1 1 1 1 .
5| . 1 1 1 1 1   5| . 1 1 1 1 1   5| . . . . . .   5| . . . . . .   5| . . . . . .
  ------------     ------------     ------------     ------------     ------------
   0 1 2 3 4 5      0 1 2 3 4 5      0 1 2 3 4 5      0 1 2 3 4 5      0 1 2 3 4 5


M 3 5 N          M 3 5 N          P 3 5 1          M 3 5 N          P 3 5 1
0| 1 1 1 1 1 .   0| 1 1 1 1 1 .   0| 1 1 1 1 1 .   0| 1 1 1 1 1 .   0| 1 1 1 1 1 .
1| 1 1 1 1 1 .   1| 1 1 1 1 1 .   1| 1 1 1 1 1 .   1| 1 1 1 1 1 .   1| 1 1 1 1 1 .
2| 1 1 1 1 1 2   2| 1 1 1 1 1 2   2| 1 1 1 1 1 2   2| 1 1 1 1 1 2   2| 1 1 1 1 1 2
3| 1 1 1 1 1 .   3| 1 1 1 1 1 .   3| 1 1 1 1 1 3   3| 1 1 1 1 1 3   3| 1 1 1 1 1 3
4| 1 1 1 1 1 .   4| 1 1 1 1 1 .   4| 1 1 1 1 1 .   4| 1 1 1 1 1 .   4| 1 1 1 1 1 .
5| . . . . . .   5| . . . . . .   5| . . . . . .   5| . . . . . .   5| . . . . . .
  ------------     ------------     ------------     ------------     ------------
   0 1 2 3 4 5      0 1 2 3 4 5      0 1 2 3 4 5      0 1 2 3 4 5      0 1 2 3 4 5
```

Example 3

```
P 0 5 2              P 0 2 2              P 0 0 2
0| . . . . . 1 1 .   0| . . 2 2 . 1 1 .   0| 3 3 2 2 . 1 1 .
1| . . . . . 1 1 .   1| . . 2 2 . 1 1 .   1| 3 3 2 2 . 1 1 .
  ----------------     ----------------     ----------------
   0 1 2 3 4 5 6 7      0 1 2 3 4 5 6 7      0 1 2 3 4 5 6 7


M 1 6 W              P 0 6 2
0| 3 3 2 2 1 1 . .   0| 3 3 2 2 1 1 4 4
1| 3 3 2 2 1 1 . .   1| 3 3 2 2 1 1 4 4
  ----------------     ----------------
   0 1 2 3 4 5 6 7      0 1 2 3 4 5 6 7
```

Public data

The public data set is intended for easier debugging and approximate program correctness checking. The public data set is stored also in the upload system and each time a student submits a solution it is run on the public dataset and the program output to stdout and stderr is available to him/her.

Link to public data set