1stNorth African Olympiad in Informatics

2025

Dox Taurus Cows

Time limit: 1 second **Memory limit: 256 MB**

Grandpa Porcellesi's old farm can be represented as an $N \times M$ grid (fenced), where each cell represents one hectare of land. Rows are numbered from 0 to $N - 1$ from top to bottom, and columns from 0 to $M - 1$ from left to right.

Porcellesi planned the fenced areas in the following way: as long as a rectangular area exists, he fences the largest possible square with its top-left corner inside the rectangle. As a result, all minimal fenced areas are squares.

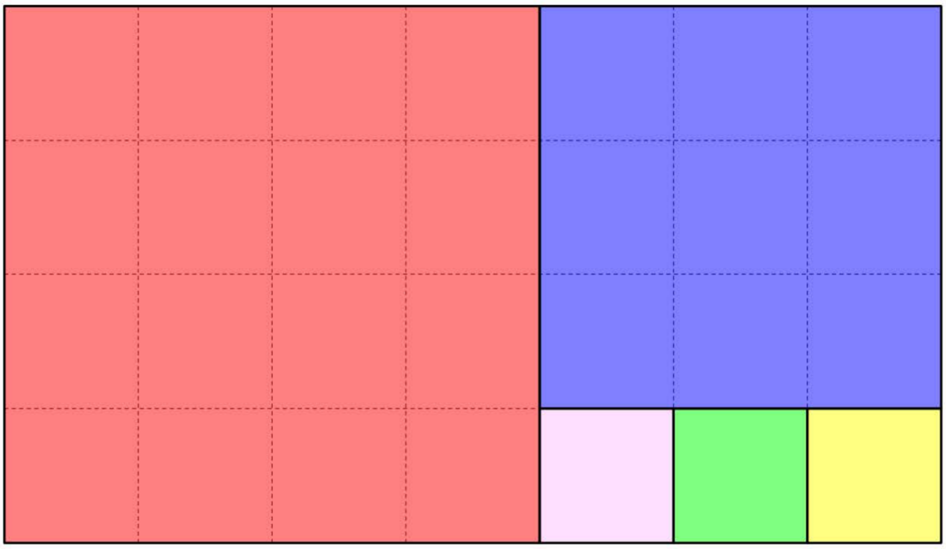


Figure 1: An example of a farm with $N = 4$ and $M = 7$.

Besides its peculiar topology, Grandpa Porcellesi's farm is inhabited by intelligent cows: the **Dox Taurus**. These cows, also known as **quantum cows**, have the ability to disappear and reappear at will. (By doing so, the cows can move between different fenced areas.)

Grandpa Porcellesi decides to monitor the cows' movements: in particular, he will record every time a cow appears or disappears from a cell in the farm.

For logistical reasons, he is interested in knowing the maximum number of cows present in a fenced area at any given time.

Problem Description

Formally, you are given Q queries of one of the following three types:

- `add (r, c)`: Adds a cow to cell (r, c)
- `remove (r, c)`: Removes a cow from cell (r, c)
- `count`: Returns the maximum number of cows in any fenced area at that moment

Help Grandpa Porcellesi answer the **Dox Taurus cows' queries**!

Input

The input consists of $Q+1$ lines:

- **Line 1**: Integers N, M, Q
- **Lines** $(1+i)$ $(1 \leq i \leq Q)$: Description of an operation:
  - `a r c`: Add a cow at (r, c)
  - `t r c`: Remove a cow from (r, c)
  - `c`: Request the maximum number of cows in a fenced square

Output

The output consists of C **lines**, where C is the number of queries of type `count`:

- **Line** i: The value returned by the i^{th} query of type `count`

Constraints

- $1 \leq N, M \leq 10^{18}$
- $0 \leq Q \leq 200,000$
- $0 \leq r < N$, $0 \leq c < M$ for each operation
- Farm is initially empty
- Every removal operation is valid (cell has at least one cow)

Subtasks

Subtask	Points	Constraints
1	0	Example cases only
2	11	$N \leq 50$, $M \leq 50$, $Q \leq 500$
3	21	$N \leq 50$, $M \leq 50$, $Q \leq 20,000$
4	20	N is a multiple of M
5	27	$Q \leq 500$
6	21	No additional constraints

Examples

Example 1

```
4 7 8
a 2 1
a 1 4
a 0 5
a 3 5
c
t 0 5
a 3 5
c
```

Output:

```
2
2
```

Example 2

```
13 9 17
a 10 5
a 11 8
c
a 9 6
c
t 10 5
c
a 11 8
a 11 8
c
t 11 8
t 11 8
c
a 9 0
a 9 4
a 10 1
c
```

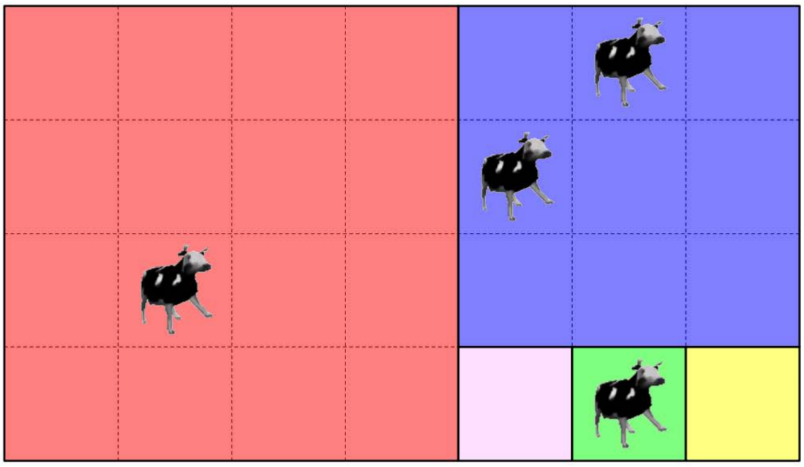
Output:

```
1
2
1
3
1
2
```

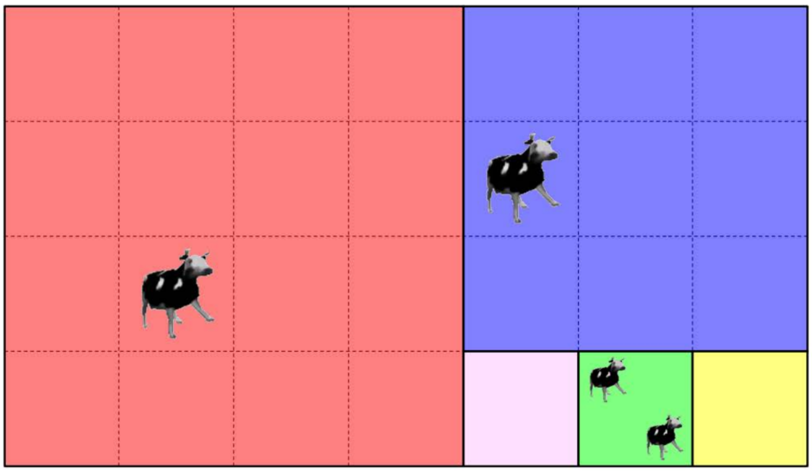
Explanation

In the first test case:

- The farm starts **empty**
- After the first 4 queries, the farm looks like this :
  - The **top-right** fenced area contains 2 **cows**.
  - the first `count` query returns 2.



- After processing the next queries, the fenced area with the most cows now has 2 **cows**.



The second `count` query also returns 2.