Farmer John operates a collection of
N farms (
$1 \leq N \leq 10^5$), conveniently numbered
$1 \ldots N$. Initially, there are no roads connecting these farms to each-other, and each farm is actively producing milk.

Due to the dynamic nature of the economy, Farmer John needs to make changes to his farms according to a series of Q update operations ($0 \leq Q \leq 2 \cdot 10^5$). Update operations come in three possible forms:

- (D x) Deactivate an active farm x, so it no longer produces milk.
- (A x y) Add a road between two active farms x and y.
- (R e) Remove the eth road that was previously added ($e = 1$ is the first road that was added).

A farm x that is actively producing milk, or that can reach another active farm via a series of roads, is called a "relevant" farm. For each farm x, please calculate the maximum i ($0 \leq i \leq Q$) such that x is relevant after the i-th update.

INPUT FORMAT (input arrives from the terminal / stdin):

The first line of input contains N and Q. The next Q lines each contain an update of one of the following forms:

```
D x
A x y
R e
```

It is guaranteed that for updates of type R, e is at most the number of roads that have been added so far, and no two updates of type R have the same value of e.

OUTPUT FORMAT (print output to the terminal / stdout):

Please output N lines, each containing an integer in the range $0 \ldots Q$.

SAMPLE INPUT:

```
5 9
A 1 2
A 2 3
D 1
D 3
A 2 4
D 2
R 2
R 1
R 3
```

SAMPLE OUTPUT:

```
7
8
6
9
9
```

In this example, roads are removed in the order $(2, 3), (1, 2), (2, 4)$.

- Farm 1 is relevant just before $(1, 2)$ is removed.
- Farm 2 is relevant just before $(2, 4)$ is removed.

- Farm 3 is relevant just before $(2, 3)$ is removed.
- Farms 4 and 5 are still active after all queries. Therefore they both stay relevant, and the output for both should be Q.

SCORING:

- Tests 2 through 5 satisfy $N \leq 10^3$, $Q \leq 2 \cdot 10^3$
- Test cases 6 through 20 satisfy no additional constraints.