Piles (Junior)

SEASON 6 – ROUND SIX – 50 points

Your good friend Ivancho has become a university student and, like all university students, keeps all of his notes in (physical) folders. Each folder contains a certain amount of papers and he has N piles, on which he piles up his folders. When he needs to take a folder from a given pile, he always takes the top folder from it. At the end of the day Ivancho wants to know how many papers there are in the top folder of each of the piles.

Input

The first row of the file `piles.in` contains an integer **N** – the number of piles on Ivanchos desk, and an integer **M** – the number of operations which he will perform (putting a folder on top of a pile or taking a folder from a pile).

M lines follow – the operation, which Ivancho performs, given in the following way:

`<operation> <pile_number> <number_of_papers_in_folder>`

Where:

- `<operation>` = **"ADD"** or **"REMOVE"**
- **0** ≤ `<pile_number>` ≤ **N-1**
- **1** ≤ `<number_of_papers_in_folder>` ≤ 2^{16}**-1** which is given only when the operation is **"ADD"**

When performing operation **"ADD"** at the top of the `<pile_number>`'th pile is placed a folder containing `<number_of_papers_in_folder>` papers.

When performing operation **"REMOVE"** the folder at the top of the `<pile_number>`'th pile is removed.

It is guaranteed that operation **"REMOVE"** is always performed on a nonempty pile.

Output

In the output file `piles.out` print N space separated integers – the number of papers in the top folder of pile number 0, 1, 2, …, N-1. If a pile has no folders, print 0.

Constraints

$3 \leq N \leq 10^3$

$5 \leq M \leq 10^4$

Time limit: 0.5 sec
Memory limit: 256 MB

Piles (Junior)

SEASON 6 – ROUND SIX – 50 points

Example test

Input (piles.in)	Output (piles.out)
3 8 ADD 0 2 ADD 0 8 ADD 2 15 REMOVE 0 REMOVE 0 REMOVE 2 ADD 0 9 ADD 2 6	9 0 6
4 6 ADD 0 4 ADD 0 15 ADD 0 2 REMOVE 0 ADD 3 14 ADD 2 2	15 0 2 14