Добрият ви приятел Иванчо вече е станал студент и сега, както всички студенти, държи всичките си записки в папки. Всяка папка съдържа определено количество листи и той има **N** купчини, на които трупа папки. Освен това, когато му се наложи да вземе папка от някоя купчина, той винаги взима най-горната от купчината. В края на деня Иванчо иска да разбере колко листи има в най-горната папка на всяка една от купчините.

**Вход**

От първия ред на файла piles.in се въвежда цяло число **N** – броя на купчините на бюрото на Иванчо, и цяло число **M** – броят на операциите, които ще извърши Иванчо (поставяне на папка върху купчина или взимане на най-горната папка от дадена купчина).

Следват **M** реда – операциите, които Иванчо извършва, зададени по следния начин:

<операция> <номер\_на\_купчина> <брой\_листи\_в\_папка>

Където:

* <операция> = **“ADD”** или **“REMOVE”**
* **0** ≤ <номер\_на\_купчина> ≤ **N-1**
* <брой\_листи\_в\_папка> присъства само когато операцията е **“ADD”**.

При операция **“ADD”** най-отгоре на купчината <номер\_на\_купчина> се поставя папка с <брой\_листи\_в\_папка> брой листи в нея.

При операция **„REMOVE”** от купчината <номер\_на\_купчина> се премахва най-горната папка.

 Входът е гарантирано, че е зададен коректно, т.е. няма да има случай, в който REMOVE се прилага върху празна купчина.

**Изход**

В изходния файл piles.out отпечатайте N цели числа, разделени с интервал – съответно броя на листите в най-горната папка на купчина номер 0, 1, 2, …, N-1. Ако една купчина е празна, отпечатайте 0.

**Ограничения**

3 ≤ *N* ≤ 103

5 ≤ *M* ≤ 104

**Ограничение за време: 0.5 сек**

**Ограничение за памет: 256 MB**

**Примерен тест**

|  |  |
| --- | --- |
| **Вход (piles.in)** | **Изход (piles.out)** |
| 3 8ADD 0 2ADD 0 8ADD 2 15REMOVE 0REMOVE 0REMOVE 2ADD 0 9ADD 2 6 | 9 0 6 |
| 4 6ADD 0 4ADD 0 15ADD 0 2REMOVE 0ADD 3 14ADD 2 2 | 15 0 2 14 |