# Задача A. Ромашки

| | |
|---|---|
| Имя входного файла: | `camomiles.in` |
| Имя выходного файла: | `camomiles.out` |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 мегабайт |

Саша любит наблюдать за ромашками. Но следить за всем полем сразу ей тяжело. Поэтому она наблюдает только за его частью, ограниченной прямоугольником размером $x \times y$. Этот прямоугольник поделён на маленькие прямоугольники размером $1 \times 1$. Во время её наблюдений могут происходить следующие события:

1. в каком-то прямоугольнике появляются или исчезают несколько ромашек;
2. к ней может заглянуть её подруга Настя и поинтересоваться, сколько видно ромашек в части поля, состоящей из нескольких прямоугольников.

## Формат входных данных

Первая строка входного файла содержит два натуральных числа $x$ и $y$ ($1 \leqslant x, y \leqslant 1\,000$). Следующая строка содержит единственное натуральное число $n$ ($1 \leqslant n \leqslant 100\,000$) — количество произошедших событий. Далее следуют $n$ записей о происходивших событиях по одной в строке. В начале строки записано число $t$. Если $t$ равно:

1, то за ним следуют 3 числа: $x_1$, $y_1$ ($1 \leqslant x_1 \leqslant x$, $1 \leqslant y_1 \leqslant y$) и $k$ ($-10\,000 \leqslant k \leqslant 10\,000$) — координаты прямоугольника и величина, на которую в нём изменилось количество видимых ромашек;

2, то за ним следуют 4 числа: $x_1$, $y_1$, $x_2$, $y_2$ ($1 \leqslant x_1 \leqslant x_2 \leqslant x$, $1 \leqslant y_1 \leqslant y_2 \leqslant y$), которые означают, что Настя попросила подсчитать количество ромашек в прямоугольниках $(x', y')$ из области: $x_1 \leqslant x' \leqslant x_2$, $y_1 \leqslant y' \leqslant y_2$.

## Формат выходных данных

Для каждого Настиного вопроса выведите искомое количество ромашек *(оно может быть отрицательным)*.

## Примеры

| camomiles.in | camomiles.out |
|---|---|
| 8 8<br>3<br>1 2 2 2<br>1 1 1 1<br>2 1 1 2 2 | 3 |

# Задача B. Вперёд!

| | |
|---|---|
| Имя входного файла: | `movetofront.in` |
| Имя выходного файла: | `movetofront.out` |
| Ограничение по времени: | 3 секунды |
| Ограничение по памяти: | 256 мегабайт |

Капрал Дукар любит раздавать приказы своей роте. Самый любимый его приказ — «Вперёд!». Капрал строит солдат в ряд и отдаёт некоторое количество приказов, каждый из которых звучит так: «Рядовые с $l_i$ по $l_j$ — вперёд!»

Перед тем, как Дукар отдал первый приказ, солдаты были пронумерованы от 1 до $n$ слева направо. Услышав приказ «Рядовые с $l_i$ по $l_j$ — вперёд!», солдаты, стоящие на местах с $l_i$ по $l_j$ включительно, продвигаются в начало ряда в том же порядке, в котором были.

Например, если в какой-то момент солдаты стоят в порядке $2, 3, 6, 1, 5, 4$, то после приказа «Рядовые с 2 по 4 — вперёд!», порядок будет таким: $3, 6, 1, 2, 5, 4$. А если потом Капрал вышлет вперёд солдат с 3 по 4, то порядок будет уже таким: $1, 2, 3, 6, 5, 4$.

Вам дана последовательность приказов Капрала. Найдите порядок, в котором будут стоять солдаты после исполнения всех приказов.

## Формат входных данных

В первой строке входного файла указаны числа $n$ и $m$ ($2 \leqslant n \leqslant 100\,000$, $1 \leqslant m \leqslant 100\,000$) — число солдат и число приказов. Следующие $m$ строк содержат приказы в виде двух целых чисел: $l_i$ и $r_i$ ($1 \leqslant l_i \leqslant r_i \leqslant n$).

## Формат выходных данных

Выведите в выходной файл $n$ целых чисел — порядок, в котором будут стоять солдаты после исполнения всех приказов.

## Примеры

| movetofront.in | movetofront.out |
|---|---|
| 6 3<br>2 4<br>3 5<br>2 2 | 1 4 5 2 3 6 |

# Задача C. И снова сумма...

| | |
|---|---|
| Имя входного файла: | `sum.in` |
| Имя выходного файла: | `sum.out` |
| Ограничение по времени: | 3 секунды |
| Ограничение по памяти: | 64 мегабайта |

Реализуйте структуру данных, которая поддерживает множество $S$ целых чисел, с которым разрешается производить следующие операции:

- $add(i)$ — добавить в множество $S$ число $i$ (если он там уже есть, то множество не меняется);
- $sum(l, r)$ — вывести сумму всех элементов $x$ из $S$, которые удовлетворяют неравенству $l \leqslant x \leqslant r$.

## Формат входных данных

Исходно множество $S$ пусто. Первая строка входного файла содержит $n$ — количество операций ($1 \leqslant n \leqslant 300\,000$). Следующие $n$ строк содержат операции. Каждая операция имеет вид либо «`+ i`», либо «`? l r`». Операция «`? l r`» задаёт запрос $sum(l, r)$.

Если операция «`+ i`» идёт во входном файле в начале или после другой операции «`+`», то она задаёт операцию $add(i)$. Если же она идёт после запроса «`?`», и результат этого запроса был $y$, то выполняется операция $add((i + y) \bmod 10^9)$.

Во всех запросах и операциях добавления параметры лежат в интервале от 0 до $10^9$.

## Формат выходных данных

Для каждого запроса выведите одно число — ответ на запрос.

## Примеры

| sum.in | sum.out |
|---|---|
| 6<br>+ 1<br>+ 3<br>+ 3<br>? 2 4<br>+ 1<br>? 2 4 | 3<br>7 |

# Задача D. Художник

| | |
|---|---|
| Имя входного файла: | `painter.in` |
| Имя выходного файла: | `painter.out` |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 мегабайт |

Не успев дорисовать свой гениальный футуристический шедевр, М. Калевич увлёкся рисованием одномерных чёрно-белых картин. Он пытается найти оптимальные местоположение и количество чёрных участков картины. Для этого он проводит на прямой белые и чёрные отрезки и после каждой из таких операций хочет знать количество чёрных отрезков на получившейся картине и их суммарную длину.

Изначально прямая белая. Ваша задача — написать программу, которая после каждой такой операции выводит в выходной файл интересующие художника данные.

## Формат входных данных

В первой строке входного файла содержится общее количество нарисованных отрезков ($1 \leqslant N \leqslant 100\,000$). В последующих $N$ строках содержится описание операций. Каждая операция описывается строкой вида $c\ x\ l$, где $c$ — цвет отрезка (‘`W`’ для белых отрезков и ‘`B`’ для чёрных), а сам отрезок имеет вид $[x; x + l]$, причём координаты обоих концов — целые числа, по модулю не превосходящие 500 000. Длина задаётся положительным целым числом.

## Формат выходных данных

После выполнения каждой из операций необходимо вывести в выходной файл на отдельной строке количество чёрных отрезков на картине и их суммарную длину, разделённые одним пробелом.

## Примеры

| painter.in | painter.out |
|---|---|
| 7<br>W 2 3<br>B 2 2<br>B 4 2<br>B 3 2<br>B 7 2<br>W 3 1<br>W 0 10 | 0 0<br>1 2<br>1 4<br>1 4<br>2 6<br>3 5<br>0 0 |