

## Задача А. Сумма на отрезке

Имя входного файла: стандартный ввод  
 Имя выходного файла: стандартный вывод  
 Ограничение по времени: 2 секунды  
 Ограничение по памяти: 256 мегабайт

Дан массив из  $N$  элементов, нужно научиться находить сумму чисел на отрезке.

### Формат входных данных

Первая строка входного файла содержит два целых числа  $N$  и  $K$  — количество чисел в массиве и количество запросов ( $1 \leq N \leq 100\,000$ ,  $0 \leq K \leq 100\,000$ ). Следующие  $K$  строк содержат следующие запросы:

1. A  $i$   $x$  — присвоить  $i$ -му элементу массива значение  $x$  ( $1 \leq i \leq n$ ,  $0 \leq x \leq 10^9$ );
2. Q  $l$   $r$  — найти сумму чисел в массиве на позициях от  $l$  до  $r$  ( $1 \leq l \leq r \leq n$ ).

Изначально в массиве живут нули.

### Формат выходных данных

На каждый запрос вида Q  $l$   $r$  нужно вывести единственное число — сумму на отрезке.

### Пример

стандартный ввод	стандартный вывод
5 9	0
A 2 2	2
A 3 1	1
A 4 2	2
Q 1 1	0
Q 2 2	5
Q 3 3	
Q 4 4	
Q 5 5	
Q 1 5	

## Задача В. Фенвик с подвохом

Имя входного файла: стандартный ввод  
 Имя выходного файла: стандартный вывод  
 Ограничение по времени: 0.4 секунд  
 Ограничение по памяти: 256 мегабайт

Вам дан массив из  $n$  элементов, изначально заполненный нулями. Вам поступают запросы двух типов:

1. прибавить ко всем числам на  $[L : R]$  число  $x$
2. узнать значение  $i$ -го элемента.

Реализуйте дерево Фенвика, которое поддерживает данные операции. Нет, мы не ошиблись. Да, это дерево Фенвика. Да, оно на это способно (а мы верим, что вы способны это придумать). Нет, слишком долго работать не будет. И да, дерево отрезков здесь не зайдет, даже не пытайтесь.

### Формат входных данных

В первой строке даны два числа  $N$  и  $M$  — количество элементов массива и количество запросов соответственно ( $1 \leq N \leq 2 \cdot 10^5, 1 \leq M \leq 6 \cdot 10^5$ )

В следующих  $m$  строках описаны запросы:

Запросы первого типа имеют вид  $+LRx$  ( $1 \leq L, R \leq N, -100 \leq x \leq 100$ ).

Запросы второго типа имеют вид  $?i$  ( $1 \leq i \leq N$ )

### Формат выходных данных

На каждый запрос второго типа выведите элемент стоящий на  $i$ -ой позиции массива.

### Пример

стандартный ввод	стандартный вывод
5 5	7
+ 1 5 7	10
? 3	7
+ 1 3 3	
? 2	
? 4	

## Задача С. Звезды

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вася любит наблюдать за звездами. Но следить за всем небом сразу ему тяжело. Поэтому он наблюдает только за частью пространства, ограниченной кубом размером  $n \times n \times n$ . Этот куб поделен на маленькие кубики размером  $1 \times 1 \times 1$ . Во время его наблюдений могут происходить следующие события:

1. В каком-то кубике появляются или исчезают несколько звезд.
2. К нему может заглянуть его друг Петя и поинтересоваться, сколько видно звезд в части пространства, состоящей из нескольких кубиков.

### Формат входных данных

Первая строка входного файла содержит натуральное число  $1 \leq n \leq 128$ . Координаты кубиков — целые числа от 0 до  $n - 1$ . Далее следуют записи о происшедших событиях по одной в строке. В начале строки записано число  $m$ . Если  $m$  равно:

- 1, то за ним следуют 4 числа —  $x, y, z$  ( $0 \leq x, y, z < N$ ) и  $k$  ( $-20000 \leq k \leq 20000$ ) — координаты кубика и величина, на которую в нем изменилось количество видимых звезд;
- 2, то за ним следуют 6 чисел —  $x_1, y_1, z_1, x_2, y_2, z_2$  ( $0 \leq x_1 \leq x_2 < N, 0 \leq y_1 \leq y_2 < N, 0 \leq z_1 \leq z_2 < N$ ), которые означают, что Петя попросил подсчитать количество звезд в кубиках  $(x, y, z)$  из области:  $x_1 \leq x \leq x_2, y_1 \leq y \leq y_2, z_1 \leq z \leq z_2$ ;
- 3, то это означает, что Васе надоело наблюдать за звездами и отвечать на вопросы Пети. Эта запись встречается во входном файле только один раз и будет последней.

Количество записей во входном файле не больше 100 002.

### Формат выходных данных

Для каждого Петинго вопроса выведите искомое количество звезд.

### Пример

стандартный ввод	стандартный вывод
2	0
2 1 1 1 1 1 1	1
1 0 0 0 1	4
1 0 1 0 3	2
2 0 0 0 0 0 0	
2 0 0 0 0 1 0	
1 0 1 0 -2	
2 0 0 0 1 1 1	
3	

## Задача D. Катый ноль

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

Реализуйте эффективную структуру данных, позволяющую изменять элементы массива и вычислять индекс  $k$ -го слева нуля на данном отрезке в массиве.

### Формат входных данных

В первой строке вводится одно натуральное число  $N$  ( $1 \leq N \leq 200\,000$ ) — количество чисел в массиве. Во второй строке вводятся  $N$  чисел от 0 до 100 000 — элементы массива. В третьей строке вводится одно натуральное число  $M$  ( $1 \leq M \leq 200\,000$ ) — количество запросов. Каждая из следующих  $M$  строк представляет собой описание запроса. Сначала вводится одна буква, кодирующая вид запроса (`s` — вычислить индекс  $k$ -го нуля, `u` — обновить значение элемента). Следом за `s` вводится три числа — левый и правый концы отрезка и число  $k$  ( $1 \leq k \leq N$ ). Следом за `u` вводятся два числа — номер элемента и его новое значение.

### Формат выходных данных

Для каждого запроса  $s$  выведите результат. Все числа выводите в одну строку через пробел. Если нужного числа нулей на запрашиваемом отрезке нет, выводите  $-1$  для данного запроса.

### Пример

<code>kthzero.in</code>	<code>kthzero.out</code>
5	4
0 0 3 0 2	
3	
u 1 5	
u 1 0	
s 1 5 3	

### Замечание

TL для Python 8 секунд