

## Задача А. Сумма на отрезке

Имя входного файла: `sum.in`  
Имя выходного файла: `sum.out`  
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:

- A  $i$   $x$  — присвоить  $i$ -му элементу массива значение  $x$  ( $1 \leq i \leq n$ ,  $0 \leq x \leq 10^9$ );
- 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.

### Примеры

<code>sum.in</code>	<code>sum.out</code>
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	

### Замечание

TL для Python 4 секунды

## Задача В. И снова сумма...

Имя входного файла: `sum2.in`  
Имя выходного файла: `sum2.out`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

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  $l \leq x \leq r$ .

### Формат входных данных

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

Если операция «+  $i$ » идет во входном файле в начале или после другой операции «+», то она задает операцию  $add(i)$ . Если же она идет 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

<code>sum2.in</code>	<code>sum2.out</code>
6	3
+ 1	7
+ 3	
+ 3	
? 2 4	
+ 1	
? 2 4	

## Задача С. Мощные юнги

Имя входного файла: `power.in`  
Имя выходного файла: `power.out`  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 256 мегабайт

Имеется список из  $n$  юнг, для каждого из которых известен его рост  $a_1, a_2, \dots, a_n$ . Рассмотрим некоторый его подсписок  $a_l, a_{l+1}, \dots, a_r$ , где  $1 \leq l \leq r \leq n$ , и для каждого натурального числа  $s$  обозначим через  $K_s$  число юнг с ростом  $s$  в этом подсписке. Назовем *мощностью* подсписка сумму произведений  $K_s \cdot K_s \cdot s$  по всем различным натуральным  $s$ . Так как количество различных чисел в массиве конечно, сумма содержит лишь конечное число ненулевых слагаемых.

Необходимо вычислить мощности каждого из  $t$  заданных подсписков.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $t$  ( $1 \leq n, t \leq 200000$ ) — длина списка и количество запросов соответственно.

Вторая строка содержит  $n$  натуральных чисел  $a_i$  ( $1 \leq a_i \leq 10^6$ ) — рост юнг.

Следующие  $t$  строк содержат по два натуральных числа  $l$  и  $r$  ( $1 \leq l \leq r \leq n$ ) — индексы левого и правого концов соответствующего подсписка.

### Формат выходных данных

Выведите  $t$  строк, где  $i$ -ая строка содержит единственное натуральное число — мощность подсписка  $i$ -го запроса.

### Примеры

<code>power.in</code>	<code>power.out</code>
3 2	3
1 2 1	6
1 2	
1 3	
8 3	20
1 1 2 2 1 3 1 1	20
2 7	20
1 6	
2 7	

## Задача D. Горилла и кладбище

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 11 секунд  
Ограничение по памяти: 1024 мегабайта

Горилле не дарили массив  $a$  из  $n$  целых неотрицательных чисел, она не покупала его, не находила на улице, а просто обнаружила его ночью на кладбище.

Горилле известно, что массивы на кладбище обычно не валяются. Вам известно, что от задачи на такую тему с такими ограничениями ничего хорошего ждать не стоит.

Даны  $q$  запросов:

- $? l r$  — вывести МЕХ подотрезка массива  $a$  от  $l$  до  $r$ ;
- $! i x$  — изменить значение в индексе  $i$  массива  $a$  на  $x$ .

МЕХ мультимножества это минимальное целое неотрицательное  $k$ , не лежащее в этом мультимножестве.

### Формат входных данных

Первая строка содержит числа  $n$  ( $1 \leq n \leq 500000$ ) и  $q$  ( $1 \leq q \leq 250000$ ).

Вторая строка содержит  $n$  чисел  $a_i$  ( $0 \leq a_i \leq n$ ) — массив  $a$ .

Каждая из следующих  $q$  строк содержит описание запроса.

Гарантируется, что запросов  $!$  не более 50000.

### Формат выходных данных

На каждый запрос  $?$  выведите запрашиваемый МЕХ в отдельной строке.

### Пример

стандартный ввод	стандартный вывод
6 8	5
4 1 0 2 2 3	0
? 1 6	3
? 4 6	4
? 2 5	0
? 2 6	5
! 3 3	
? 1 6	
! 4 0	
? 1 6	

### Замечание

В примере запросы выглядят следующим образом:

- Изначально массив равен  $[4, 1, 0, 2, 2, 3]$
- В первом запросе считается МЕХ от  $4, 1, 0, 2, 2, 3$  и он равен 5.
- Во втором запросе считается МЕХ от  $2, 2, 3$  и он равен 0.
- В третьем запросе считается МЕХ от  $1, 0, 2, 2$  и он равен 3.
- В четвёртом запросе считается МЕХ от  $1, 0, 2, 2, 3$  и он равен 4.
- Пятый запрос меняет массив. Теперь он равен  $[4, 1, 3, 2, 2, 3]$ .
- В шестом запросе считается МЕХ от всего массива и он равен 0.
- Седьмой запрос меняет массив. Теперь он равен  $[4, 1, 3, 0, 2, 3]$ .
- В восьмом запросе снова считается МЕХ от всего массива и теперь он равен 5.

## Задача Е. Сычи

Имя входного файла: owls.in  
Имя выходного файла: owls.out  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 256 мегабайт

Мурка, не ходи, там сыч  
На подушке вышит,  
Мурка серый, не мурлычь,  
Дедушка услышит.  
Няня, не горит свеча,  
И скребутся мыши.  
Я боюсь того сыча,  
Для чего он вышит?

---

Анна Ахматова

Есть  $n$  сычей. Известны пары друзей. Три сыча образуют компанию, если все три попарно дружат. Требуется найти количество компаний сычей.

### Формат входных данных

Первая строка входных данных содержит два целых числа  $n$  и  $m$  — количество сычей и дружеских связей соответственно ( $1 \leq n, m \leq 3 \cdot 10^5$ ). Каждая из следующих  $m$  строк содержит по два целых числа от 1 до  $n$  — номера сычей, которые дружат. Гарантируется, что никакая пара не указана два раза и сыч не дружит сам с собой.

### Формат выходных данных

В единственной строке выходного файла выведите количество компаний сычей.

### Примеры

owls.in	owls.out
6 6 1 2 2 3 3 1 4 2 3 4 5 1	2

## Задача F. Лунки

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

Маленький Петя очень любит играть. Больше всего он любит играть в игру «Лунки». Это игра для одного игрока со следующими правилами:

Есть  $N$  лунок, расположенных в ряд, пронумерованных слева направо числами от 1 до  $N$ . У каждой лунки изначально установлена своя сила выброса (у лунки с номером  $i$  она равна  $a_i$ ). Если вбросить шарик в лунку  $i$ , то он тут же вылетит из нее и попадет в лунку  $i + a_i$ , после чего он опять вылетит и т.д.. Если же лунки с таким номером нету, то он просто вылетит за край ряда. На каждом из  $M$  ходов игрок выбирает одно из двух действий:

- Установить силу выброса лунки  $a$  равной  $b$ .
- Вбросить шарик в лунку  $a$  и посчитать количество прыжков шарика, прежде чем он вылетит за край ряда, а так же записать номер лунки, после выпрыгивания из которой шарик вылетел за край.

У Пети есть некоторые проблемы с математикой, поэтому, как Вы уже догадались, именно Вам предстоит произвести все подсчеты.

### Формат входных данных

Первая строка содержит два числа  $N$  и  $M$  ( $1 \leq N \leq 10^5$ ,  $1 \leq M \leq 10^5$ ) — количество лунок в ряду и количество ходов. Следующая строка содержит  $N$  целых положительных чисел, не превышающих  $N$  — начальные силы выброса лунок. Следующие  $M$  строк задают ходы, сделанные Петей. Каждая строка может быть двух типов:

- 0 a b
- 1 a

Тут, первый тип означает что требуется установить силу выброса лунки  $a$  равной  $b$ , а второй означает что требуется вбросить мячик в лунку с номером  $a$ . Числа  $a$  и  $b$  — целые положительные и не превышают  $N$ .

### Формат выходных данных

Для каждого хода второго типа (задающего вбрасывание шарика) в порядке следования во входном файле выведите два числа через пробел в отдельной строке — номер последней лунки перед вылетом шарика за край и количество прыжков.

### Примеры

holes.in	holes.out
8 5	8 7
1 1 1 1 1 2 8 2	8 5
1 1	7 3
0 1 3	
1 1	
0 3 4	
1 2	

## Задача G. Фаброзавры-дизайнеры

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

Фаброзавры известны своим тонким художественным вкусом и увлечением ландшафтным дизайном. Они живут около очень живописной реки и то и дело перестраивают тропинку, идущую вдоль реки: либо насыпают дополнительной земли, либо срывают то, что есть. Для того, чтобы упростить эти работы, они поделили всю тропинку на горизонтальные участки, пронумерованные от 1 до  $N$ , и их переделки устроены всегда одинаково: они выбирают часть дороги от  $L$ -ого до  $R$ -ого участка (включительно) и изменяют (увеличивают или уменьшают) высоту на всех этих участках на одну и ту же величину (если до начала переделки высоты были разными, то и после переделки они останутся разными).

Поскольку, как уже говорилось, у фаброзавров тонкий художественный вкус, каждый из них считает, что их река лучше всего выглядит с определенной высоты. Поэтому им хочется знать, есть ли поблизости от их дома место на тропинке, где высота на их взгляд оптимальна. Помогите им в этом разобраться.

### Формат входных данных

Первая строка входного файла содержит два числа  $N$  и  $M$  — длину дороги и количество запросов соответственно ( $1 \leq N, M \leq 10^5$ ). На второй строке содержатся  $N$  чисел, разделенных пробелами — начальные высоты соответствующих частей дороги; высоты не превосходят  $10^4$  по модулю. В следующих  $M$  строках содержатся запросы по одному на строке.

Запрос  $+ L R X$  означает, что высоту частей дороги от  $L$ -ой до  $R$ -ой (включительно) нужно изменить на  $X$ . При этом  $1 \leq L \leq R \leq N$ , а  $|X| \leq 10^4$ .

Запрос  $? L R X$  означает, что нужно проверить, есть ли между  $L$ -ым и  $R$ -ым участками (включая эти участки) участок, где дорога проходит точно на высоте  $X$ . Гарантируется, что  $1 \leq L \leq R \leq N$ , а  $|X| \leq 10^9$ .

### Формат выходных данных

На каждый запрос второго типа нужно вывести в выходной файл на отдельной строке одно слово «YES» (без кавычек), если нужный участок существует, и «NO» в противном случае.

### Примеры

<code>fabro.in</code>	<code>fabro.out</code>
10 5	NO
0 1 1 3 3 3 2 0 0 1	YES
? 3 5 2	YES
+ 1 4 1	
? 3 5 2	
+ 7 10 2	
? 9 10 3	

## Задача Н. Опять $k$ -я статистика

Имя входного файла: `kthstat.in`  
Имя выходного файла: `kthstat.out`  
Ограничение по времени: 10 секунд  
Ограничение по памяти: 256 мегабайт

Изначально вам дан массив целых чисел.

Нужно уметь отвечать на три запроса:

- $+ i x$  — Вставить на  $i$ -ю позицию число  $x$  (размер массива увеличивается на 1)
- $- i$  — Удалить число на  $i$ -й позиции (размер массива уменьшается на 1)
- $? L R x$  — Сказать, сколько чисел  $y$  на позициях  $L \leq i \leq R$  таких, что  $y \leq x$  ( $|x| \leq 10^9$ )

Все индексы  $i, L, R$  нумеруются с нуля. Все числа в запросах целые. Все запросы корректны. Пример запроса: “+ 0  $x$ ” означает “добавление  $x$  в начало массива”. Исходное число элементов в массиве —  $0 \leq N \leq 10^5$ , числа в массиве по модулю не превышают  $10^9$ . Число запросов —  $1 \leq K \leq 10^5$ .

### Формат входных данных

Первое число  $n$  — размер массива. Затем  $n$  целых чисел — сам массив. Далее следуют много запросов... Читайте пока читается :)

### Формат выходных данных

Ну понятно же что выводить, ну... Ответьте на запросы :)

### Примеры

<code>kthstat.in</code>	<code>kthstat.out</code>
3 0 1 0 ? 0 1 0 + 0 1 ? 0 3 0	1 2
5 5 4 4 0 5 ? 0 0 3 + 5 1 - 4 ? 0 3 0 + 2 0	0 1
10 1 4 5 0 1 1 7 7 1 7 - 9 + 1 1 ? 0 3 6 ? 2 2 4 - 9 ? 4 5 1 ? 1 2 2 + 9 8 ? 1 6 2 + 5 1	4 1 2 1 4

### Замечание

Pa pa do do do do vap Pi pi adi de di de waa Wi pi pa pa pa do baby