

Задача А. Фаброзавры-дизайнеры

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

Фаброзавры известны своим тонким художественным вкусом и увлечением ландшафтными 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 входного файла

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 выходного файла

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 В. Танцующие слоны

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

Танцующие слоны — это зрелищное шоу в Паттайе, в котором участвуют N слонов, танцующих на одной линии, называемой сценой.

В результате многолетних тренировок слоны, участвующие в шоу, разучили большое количество танцевальных движений. Все шоу состоит из последовательности актов. В каждом акте только один слон совершает одно танцевальное движение, в результате которого он может переместиться на другую позицию на сцене.

Постановщики шоу хотят сделать фотоальбом, который бы содержал фотографии всего шоу. После каждого акта они хотят сделать фотографии всех слонов.

В любой момент времени на протяжении шоу некоторое количество слонов может находиться в одной и той же позиции — это значит, что слоны просто стоят рядом. Одна фотокамера может фотографировать группу слонов тогда и только тогда, когда все позиции, в которых находятся слоны, лежат на отрезке длины L (обе границы отрезка включаются в него). Так как слоны могут располагаться вдоль всей сцены, то может потребоваться несколько фотокамер, чтобы сфотографировать всех слонов одновременно. К примеру, предположим, что $L = 10$ и слоны располагаются на сцене в позициях 10, 15, 17, и 20 соответственно. В этот момент достаточно одной фотокамеры, чтобы сфотографировать всех слонов, как это показано ниже. (Слоны изображены как треугольники; фотокамеры изображены как трапеции).

В последующем акте слон, находящийся в позиции 15, в результате танцевального движения перемещается в позицию 32. После этого акта необходимо уже не менее двух фотокамер для того, чтобы сфотографировать всех слонов одновременно.

В следующем акте слон, находящийся в позиции 10, перемещается в позицию 7. В данном случае понадобится 3 фотокамеры для того, чтобы сфотографировать всех слонов.

В данной задаче вы должны определить минимальное количество фотокамер, необходимых для того, чтобы сделать фотографии после каждого акта шоу. Следует отметить, что количество необходимых фотокамер может увеличиваться, уменьшаться, или оставаться тем же самым от акта к акту.

Формат входного файла

Строка 1 содержит N , L , и M , где M — количество актов в шоу.

Строки от 2 до $(N+1)$ содержат начальные позиции; то есть строка с номером $(k+2)$ содержит значение $X[k]$ для каждого $0 \leq k < N$.

Строки от $(N+2)$ до $(N+M+1)$ содержат информацию об M актах; то есть, строка с номером $(N+1+j)$ для $1 \leq j \leq M$ содержит значения $i[j]$, $y[j]$, разделенные одним пробелом, и обозначает, что в j -м акте слон с номером $i[j]$ перемещается в позицию $y[j]$.

Формат выходного файла

Для каждого запроса выведите одно число — ответ на задачу.

Примеры

elephants.in	elephants.out
4 10 5	1
10	2
15	2
17	2
20	3
2 16	
1 25	
3 35	
0 38	
2 0	

Задача С. Суровый корректор

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

По мнению Александра Павловича, текст необычайно красив, если некоторые *особые* слова (например, «коммунизм», «Ленин», «счастье») встречаются не слишком часто, но и не слишком редко, к тому же достаточно равномерно. Александр Павлович работает корректором. К нему поступают тексты, он имеет право их некоторым образом менять, после чего возвращает уже исправленную версию. В связи со своими воззрениями о красоте Александру Павловичу постоянно приходится проверять, сколько *особых* слов сейчас в той или иной части текста. Он настолько устал от рутинного подсчёта: «а сколько тут особых слов?», «а сколько тут?», что просит вас помочь ему автоматизировать этот процесс.

Формат входного файла

Первая строка входного файла содержит текст, в котором ищутся особые слова. Следующая строка содержит N ($1 \leq N \leq 10^5$) — количество *особых* слов. Следующие N строк содержат *особые* слова. Все *особые* слова различны. Суммарная длина строк не превосходит 10^5 . В следующей строке дано Q ($1 \leq Q \leq 10^5$) — количество интересных Александру Павловичу отрезков. Следующие Q строк содержат сами отрезки.

Формат выходного файла

Выведите Q чисел — количества вхождений особых слов в соответствующий отрезок текста.

Примеры

<code>corrector.in</code>	<code>corrector.out</code>
abacababa	5 0 2
2	
a	
aba	
3	
5 9	
2 2	
2 6	

Задача D. Переселение

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

В общежитии, где разместили преподавателей летних сборов по чёрной магии и колдовству, проводится ремонт. Поэтому время от времени людям, проживающим в какой-то комнате, приходится переселяться в какую-то другую комнату.

Для переселения разработан план — список троек чисел (T_i, A_i, B_i) : вечером дня T_i все жители комнаты A_i переселяются в комнату B_i . Жители всех остальных комнат, в том числе B_i , при этом остаются на своих местах.

Но не всё так просто: иногда в план вносятся изменения. Каждое изменение — это добавление тройки (T_i, A_i, B_i) . Известно, что до и после каждого изменения все T_i , присутствующие в плане, различны.

К администрации общежития иногда поступают вопросы: в какой комнате будет утром дня R_i по плану жить человек, если он заселяется утром дня L_i в комнату S_i . Реализуйте программу, которая отвечает на такие вопросы.

Формат входного файла

В первой строке задано два числа n и m ($1 \leq n, m \leq 10^5$) — количество комнат и начальное количество перемещений. Следующие m строк содержат по три числа T_i, A_i, B_i .

Следующая строка задает число k ($1 \leq k \leq 10^5$) — количество запросов. Следующие k строк задают запросы одного из двух видов

- `? $t_1 t_2 a$` — сказать, где будет находиться утром дня номер t_2 человек, заселившийся утром дня номер t_1 в комнату a .
- `+ tab` — добавить в план переселения операцию перемещения вечером дня номер t всех жильцов комнаты a в комнату b .

Комнаты нумеруются от 0 до $n - 1$, индексы массивов от 0. Номера дней — целые числа от 0 до 10^9 .

Формат выходного файла

Для каждого запроса выведите одно число — ответ на задачу.

Примеры

<code>blocks.in</code>	<code>blocks.out</code>
3 3	1
10 0 1	1
20 0 2	0
30 2 1	2
11	2
? 10 20 0	0
? 10 21 0	2
? 11 20 0	0
? 11 21 0	0
? 20 21 0	1
+ 25 2 0	
? 21 25 0	
? 20 25 0	
? 20 26 0	
? 20 26 2	
? 0 1000000000 1	