

## Задача А. Освещение площади

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

В городе построили новую прямоугольную площадь размером  $m$  на  $n$  метров. Для освещения площади мэр хочет заказать инновационные фонари, каждый из которых освещает квадрат  $k \times k$  метров, стороны которого параллельны границам площади.

Мэр не хочет потратить весь бюджет города, поэтому он хочет купить как можно меньше фонарей, чтобы осветить всю площадь. Помогите ему понять, сколько фонарей надо купить.

### Формат входных данных

На ввод подается три целых числа  $n$ ,  $m$  и  $k$  ( $1 \leq n, m, k \leq 10^9$ ) — длина площади, ширина площади и длина стороны квадрата, который освещает фонарь.

### Формат выходных данных

Выведите минимальное количество фонарей, которое требуется для освещения всей площади.

### Примеры

стандартный ввод	стандартный вывод
10 9 3	12
4 6 2	6

## Задача В. Морской бой

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Рассмотрим классическую игру в «Морской бой».

Согласно правилам, у каждого игрока имеется поле размера  $10 \times 10$  клеток, на котором он должен расставить 10 кораблей: один «четырёхпалубник» (занимает  $1 \times 4$  клетки), два «трехпалубника» (размера  $1 \times 3$  клетки), три «двухпалубника» (размера  $1 \times 2$  клетки) и четыре «однопалубника» (имеют размер  $1 \times 1$ ). При этом должны быть выполнены следующие условия:

- на поле должны быть выставлены все корабли;
- каждый корабль целиком помещается внутрь поля;
- множество клеток, занимаемых каждым кораблем, образует прямоугольник соответствующего размера;
- каждый корабль расположен либо вертикально, либо горизонтально;
- любые две клетки, разных кораблей, не могут совпадать, а также касаться друг друга по стороне или углу.

Будем описывать расположение кораблей с помощью таблицы  $10 \times 10$ , где в каждом элементе находится символ '#', если соответствующая клетка занята кораблем, и '.' в противном случае. Ваша задача — по заданному полю  $10 \times 10$  определить, соответствует ли оно корректной расстановке кораблей, согласующейся с правилами «Морского боя».

### Формат входных данных

Ввод состоит из 10 строк, разделенных переводом строки, по 10 символов в каждой — описание поля. Гарантируется, что каждый символ поля равен либо '#', либо '.'.

### Формат выходных данных

Выведите «YES», если описанное во входных данных поле соответствует корректной расстановке кораблей в игре «Морской бой», и «NO» иначе.

### Примеры

стандартный ввод	стандартный вывод
<pre>##### .....##. .#.#.... .#..... .#.##.... ..... ####...#.. .....#.. .....#.. ##.....</pre>	YES
<pre>##.#.... .....##. .#.#....# .#..... .#..... ..... ####...#.. .....#.. .....#.. ##.....#</pre>	NO

## Задача С. Витрина ковров

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Рене — дизайнер абстракционистских ковров. Ковры, которые создаёт Рене, представляют собой прямоугольники  $n \times t$ , где  $n$  и  $t$  — натуральные числа. Каждый ковер разделен на  $n \times t$  одинаковых единичных квадратов.

Рене оформляет витрину ковров на мебельной выставке. Витрина представляет собой прямоугольную площадку со сторонами  $h \times w$ , где  $h$  и  $w$  — натуральные числа, пол на витрине также разделен на единичные квадраты.

Рене хочет, чтобы ковры были выложены на витрине аккуратно.

Рене считает, что ковры выложены аккуратно, если выполняются следующие правила. Каждый ковер может лежать на полу, либо на другом ковре, причем каждый единичный квадрат ковра должен находиться строго над единичным квадратом пола или ковра под ним. При этом если один ковер лежит на другом ковре, то он должен целиком находиться внутри него и иметь строго меньшую площадь.

Рене хочет выложить на витрине ковры максимальной суммарной площади. Помогите посчитать, какую максимальную суммарную площадь ковров Рене может представить на выставке, чтобы их можно было аккуратно выложить на витрине.

### Формат входных данных

В единственной строке ввода даны два натуральных числа  $h$  и  $w$  — размеры площадки, предназначенной для витрины ковров ( $1 \leq h, w \leq 10^6$ ).

### Формат выходных данных

Выведите единственное целое число — какую максимальную суммарную площадь ковров можно будет аккуратно уложить на витрине.

### Примеры

стандартный ввод	стандартный вывод
1 2	4
2 2	12
3 2	22

## Задача D. Перерисованный граф

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

На одном очень скучном уроке по математике Маша нарисовала у себя в тетради неориентированный граф с  $n$  вершинами и  $m$  ребрами. Вершины графа были пронумерованы от 1 до  $n$ .

После урока она ушла на перемену, а когда вернулась, то обнаружила, что граф, кажется, изменился!

Машин сосед по парте, Паша, признался, что в её отсутствие он немного поменял граф. А именно, он выполнял следующую операцию: взять три различных вершины графа,  $a$ ,  $b$  и  $c$ , и после этого для каждой пары вершин  $(a, b)$ ,  $(b, c)$  и  $(a, c)$  изменить соответствующее ребро: если оно было в текущем графе, удалить его, а если его не было — провести.

Эту операцию он совершил не один, а несколько (возможно, ноль или один) раз, каждый раз заново выбирая  $a$ ,  $b$  и  $c$ .

Маша хочет проверить слова своего соседа, поэтому просит вас найти любую последовательность действий Паши, которая могла бы превратить исходный граф в тот, что сейчас у Маши в тетради.

### Формат входных данных

В первой строке заданы три числа —  $n$ ,  $m$  и  $k$  ( $3 \leq n \leq 10^5$ ;  $0 \leq m, k \leq 10^5$ ) — количество вершин, количество ребер в исходном графе, и количество ребер в получившемся графе, соответственно.

Каждая из следующих  $m$  строк содержит два числа,  $u_i$  и  $v_i$  ( $1 \leq u_i, v_i \leq n$ ;  $u_i \neq v_i$ ), обозначающие вершины, соединенные  $i$ -м ребром в исходном графе.

Следующие  $k$  строк содержат описание получившегося графа в том же формате. Гарантируется, что оба графа не содержат петель и кратных ребер.

### Формат выходных данных

Если действиями Паши невозможно получить второй граф из первого, в единственной строке выведите «NO».

Иначе, в первой строке выведите «YES». В следующей строке выведите количество операций  $t$  ( $0 \leq t \leq 2 \cdot 10^5$ ), которое необходимо было совершить Паше. В каждой из следующих  $t$  строк выведите по три числа —  $a_i$ ,  $b_i$ ,  $c_i$  ( $1 \leq a_i, b_i, c_i \leq n$ , числа  $a_i$ ,  $b_i$ ,  $c_i$  попарно различны), обозначающих тройку вершин, которую должен был выбрать Паша на  $i$ -м шаге.

Обратите внимание, что вам не требуется минимизировать количество действий Паши, однако необходимо, чтобы оно не превосходило  $2 \cdot 10^5$ . Можно показать, что если решение существует, то существует и решение, состоящее из не более чем  $2 \cdot 10^5$  действий.

### Примеры

стандартный ввод	стандартный вывод
3 0 3 1 2 2 3 3 1	YES 1 1 2 3
4 3 3 1 2 2 3 3 4 1 3 4 2 2 3	YES 2 1 3 4 1 2 4
3 1 1 1 2 2 3	NO

## Задача Е. Переполох в бухгалтерии

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

В бухгалтерии гранд отеля переполох: один из гостей — Сергей, куда-то пропал и не выходит на связь, а его счет за проживание не оплачен!

Бухгалтеры гранд отеля очень ответственно относятся к своим обязанностям, поэтому для каждого посетителя заводится отдельный журнал, в котором есть 2 колонки: «услуга» и «стоимость». В графе «услуга» каждый день записываются все предоставленные отелем удобства: завтрак в постель, такси, выдача канцелярии и прочее, туда же входит и проживание в номере, которое отдельно указывается за каждый день. В графе «стоимость» напротив каждой услуги записана ее стоимость в местной валюте. Стоимость одной и той же услуги, в том числе проживания, не меняется в процессе проживания гостя.

К сожалению, из-за суматохи, вызванной исчезновением Сергея, на первую колонку журнала один из работников отеля случайно пролил чернила. По оставшейся информации известно, что Сергей жил в гранд отеле  $n$  дней, а также известны стоимости  $m$  услуг, предоставленных ему за все эти дни.

Поскольку Сергей — очень важный гость, у него был свой, нестандартный тариф на проживание, а чему он был равен — никто не помнит. Бухгалтерский отдел теперь хочет узнать, чему могла быть равна стоимость проживания Сергея в отеле за день.

### Формат входных данных

В первой строке даны два целых числа  $n, m$  ( $1 \leq n \leq m \leq 3 \cdot 10^5$ ) — количество дней проживания Сергея в отеле и количество записей об услугах.

Во второй строке даны  $m$  целых чисел  $c_1, c_2, \dots, c_m$  ( $1 \leq c_i \leq 10^9$ ) — стоимости услуг за все дни.

### Формат выходных данных

В первой строке выведите единственное число  $k$  — количество возможных вариантов стоимости номера за день. Гарантируется, что  $k \neq 0$ .

Во второй строке выведите  $k$  чисел — сами возможные варианты стоимости в произвольном порядке.

### Пример

стандартный ввод	стандартный вывод
2 10	3
1 3 6 5 3 2 4 5 3 2	3 5 2

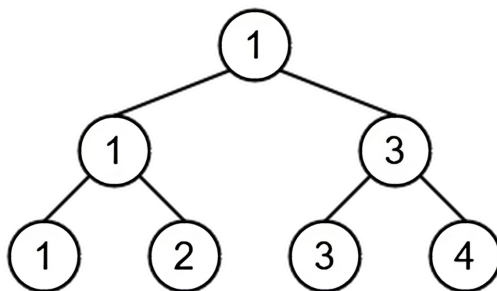
## Задача F. Дерево отрезков

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Лиза очень любит деревья отрезков. Сегодня, найдя дома массив различных натуральных чисел длины  $2^k$ , Лиза незамедлительно построила из него дерево отрезков с операцией минимума.

Дерево отрезков с операцией минимума — это двоичное дерево, у каждой вершины есть левый и правый сын. Листья дерева, если рассматривать их слева направо, содержат элементы исходного массива, а значения в других вершинах вычисляется, как минимум из значений в ее детях.

Например, если исходный массив равен  $[1, 2, 3, 4]$ , то дерево отрезков с операцией минимума будет выглядеть так:



Лиза попросила свою подругу Машу нарисовать построенное дерево на листочке, что Маша и сделала. На следующий день Лиза попыталась вспомнить, как выглядел исходный массив, но найдя листочек, с удивлением обнаружила, что Маша просто выписала на него значения в вершинах дерева в случайном порядке.

Помогите Лизе восстановить исходный массив, либо выяснить, что Маша неправильно записала значения.

### Формат входных данных

В первой строке дано одно число  $n$  — количество элементов дерева отрезков, записанных на листочке ( $1 \leq n \leq 2 \cdot 10^5$ ). Гарантируется, что существует целое число  $k$ , такое что  $n = 2^{k+1} - 1$ .

Во второй строке содержатся  $n$  чисел  $a_1, a_2, \dots, a_n$  — элементы дерева отрезков, записанные на листочке ( $1 \leq a_i \leq 10^9$ ).

### Формат выходных данных

Выведите  $-1$ , если из записанных на листочке чисел невозможно восстановить дерево отрезков с операцией минимума, построенное из массива различных натуральных чисел. Иначе исходный массив. Если существует несколько решений, разрешается вывести любое.

### Примеры

стандартный ввод	стандартный вывод
7 2 3 1 1 4 1 3	1 2 3 4
3 1 2 3	-1
3 1 1 1	-1

### Замечание

В третьем тестовом примере из набора чисел можно построить дерево отрезков с операцией минимума, однако, в изначальном массиве в таком случае числа не будут различны.

## Задача G. Подъем на лифте

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Каждый раз когда Катя заходит в лифт, в зеркале она видит отражение номера этажа, на котором она зашла.

Номер этажа показывает табло, которое отображает цифры так, как показано на картинке.



Из-за того, что номер этажа виден в зеркале, его изображение получается отраженным относительно вертикальной оси: порядок цифр меняется на противоположный и цифры выглядят как свои отражения. При этом если цифра  $x$  после отражения полностью совпадает с другой цифрой  $y$ , то мозг воспринимает ее как  $y$ , а вот если она не совпадает ни с какой другой цифрой, то, несмотря на то, что она отражена, мозг воспринимает  $x$  как  $x$ .

Какое число увидит Катя, когда зайдет в лифт, если известно, что она живет на  $k$ -м этаже.

### Формат входных данных

Дано целое число  $k$  ( $1 \leq k \leq 10^{18}$ ) — этаж, на котором живет Катя.

### Формат выходных данных

Выведите единственное число, которое Катя увидит в зеркале, без ведущих нулей.

### Примеры

стандартный ввод	стандартный вывод
13	31
250	25
1234567890	987624351

## Задача Н. Юрик и важные дела

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Юрик — очень занятой человек, поэтому он не успевает выполнять все необходимые дела вовремя. К концу недели у него скопилось  $n$  невыполненных дел. Для удобства пронумеруем их целыми числами от 1 до  $n$ . Сначала Юрик решил, что на выходных будет выполнять дела в порядке их нумерации. Однако, он быстро понял, что некоторые дела важнее других, поэтому решил изменить порядок их выполнения.

У Юрика есть любимая *перестановка*  $p_1, p_2, \dots, p_n$ , и он решил воспользоваться ей, чтобы изменить порядок выполнения дел. Помимо того, что Юрик очень занятой человек, он также очень непостоянный, поэтому он решил изменить порядок выполнения дел  $q$  раз.

Изначально Юрик записал на лист бумаги дела в том порядке, в котором он их будет выполнять:  $1, 2, 3, \dots, n$ . После этого он  $q$  раз выберет некоторые два числа  $l$  и  $r$  ( $1 \leq l \leq r \leq n$ ), после чего изменит порядок дел, которые на данный момент находятся в списке на позициях с номерами от  $l$  до  $r$ .

Изменение порядка выполнения дел происходит следующим образом. Сначала Юрик назначает делам, которые находятся в списке на позициях с номерами от  $l$  до  $r$  *приоритеты*. Делу на позиции  $l$  будет назначен приоритет  $p_1$ , делу на позиции  $l + 1$  — приоритет  $p_2$ , и так далее. Таким образом, делу на позиции  $r$  будет назначен приоритет  $p_{r-l+1}$ . После этого Юрик переупорядочит данные дела в порядке **возрастания** их приоритетов. Для лучшего понимания процесса обратите внимание на пояснение к примерам.

После того, как Юрик изменил порядок выполнения дел целых  $q$  раз, он окончательно запутался, и теперь хочет понять, какое дело он выполнит  $k$ -м по очереди. Помогите ему это выяснить.

Напомним, что перестановкой  $p_1, p_2, \dots, p_n$  называется массив попарно различных чисел от 1 до  $n$ .

### Формат входных данных

Первая строка содержит три целых числа  $n$ ,  $q$  и  $k$  ( $1 \leq n \leq 100\,000$ ;  $1 \leq q \leq 100\,000$ ;  $1 \leq k \leq n$ ).

Вторая строка содержит  $n$  целых чисел  $p_1, p_2, \dots, p_n$  ( $1 \leq p_i \leq n$ ) — любимую перестановку Юрика.

Каждая из следующих  $q$  строк содержит два целых числа  $l_i$  и  $r_i$  ( $1 \leq l_i \leq r_i \leq n$ ) — начало и конец отрезка дел, порядок которых Юрик изменит в  $i$ -й раз.

### Формат выходных данных

Выведите одно целое число  $t$  ( $1 \leq t \leq n$ ) — номер дела, которое Юрик выполнит  $k$ -м по счету после всех изменений порядка выполнения дел.

### Пример

стандартный ввод	стандартный вывод
8 6 3 1 5 2 8 7 4 3 6 1 8 2 4 7 8 1 1 1 3 2 7	7

### Замечание

Рассмотрим первый пример. Изначально Юрик планировал выполнять дела в следующем порядке: 1, 2, 3, 4, 5, 6, 7, 8. Однако, он решил изменить порядок выполнения дел 6 раз.

В первый раз Юрик выбрал дела, находящиеся в списке на позициях от 1 до 8 (то есть все имеющиеся у него дела). Первому делу будет назначен приоритет 1, второму делу — приоритет 5,



третьему делу — приоритет 2, и так далее. После этого Юрик упорядочит дела в порядке возрастания приоритетов, поэтому получится последовательность: 1, 3, 7, 6, 2, 8, 5, 4.

Во второй раз Юрик выбрал дела, находящиеся в списке на позициях от 2 до 4: 3, 7 и 6. Делу с номером 3 был назначен приоритет 1, делу с номером 7 — приоритет 5, а делу с номером 6 — приоритет 2. После того, как Юрик упорядочит дела в порядке возрастания приоритетов, получится последовательность 1, 3, 6, 7, 2, 8, 5, 4.

В третий раз Юрик выбрал последние два дела в текущем порядке: 5 и 4. Первому из них был назначен приоритет 1, а второму — приоритет 5, поэтому порядок их выполнения не изменился.

В четвертый раз было выбрано одно первое дело.

В пятый раз были выбраны первые три дела в текущем порядке, и после назначения приоритетов дела 3 и 6 поменялись местами.

Наконец, в шестой раз были выбраны все дела, кроме первого и последнего. Итоговый порядок выполнения дел выглядит следующим образом: 1, 6, 7, 5, 3, 8, 2, 4.

## Задача I. Крыши

Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Во время археологических раскопок были обнаружены руины античного храма. Лучше всего сохранилась колоннада, состоящая из  $n$  колонн, расположенных в ряд. Все колонны частично разрушились, поэтому оказалось, что высоты всех колонн различны. Высота  $i$ -й колонны равна  $h_i$ .

Чтобы предотвратить дальнейшее разрушение колонн из-за непогоды, было принято решение накрыть их крышами. Каждая крыша располагается горизонтально и одним концом прикрепляется к верху некоторой колонны. Можно установить крышу, которая накрывает отрезок колонн с  $i$ -й по  $j$ -ю ( $1 \leq i \leq j \leq n$ ), если выполнено одно из следующих условий:

- Если  $i$ -я колонна выше всех остальных на отрезке  $[i, j]$ , то можно закрепить крышу левым концом на  $i$ -й колонне.
- Если  $j$ -я колонна выше всех остальных на отрезке  $[i, j]$ , то можно закрепить крышу правым концом на  $j$ -й колонне.

Наверху каждой колонны можно закрепить не более одной крыши. Стоимость закрепления крыши на  $i$ -й колонне равна  $c_i$  вне зависимости от того, направлена она влево или вправо и сколько колонн накрывает.

Требуется закрепить крыши на некоторых колоннах таким образом, чтобы каждая колонна была накрыта хотя бы одной крышей, и суммарная стоимость была минимальна.

### Формат входных данных

В первой строке находится число  $n$  ( $1 \leq n \leq 200\,000$ ) — количество колонн.

Во второй строке даны  $n$  целых чисел  $h_1, h_2, \dots, h_n$  ( $1 \leq h_i \leq 10^9$ ) — высоты колонн. Гарантируется, что все  $h_i$  различны.

В третьей строке даны  $n$  целых чисел  $c_1, c_2, \dots, c_n$  ( $1 \leq c_i \leq 10^9$ ) — стоимости закрепления крыш на колоннах.

### Формат выходных данных

Выведите одно число — минимальную стоимость накрыть все колонны крышами.

### Примеры

стандартный ввод	стандартный вывод
3 3 10 7 2 5 2	7
1 5 2	2

## Задача J. Побег слайма

Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Никите на день рождения подарили набор для создания слаймов. Обрадовавшись подарку, Никита сделал квадратного слайма, и положил его на стол. Слайм, испугавшись предстоящих экспериментов Никиты, решил от него сбежать.

Стол представляет собой прямоугольное поле из  $n$  строк и  $m$  столбцов. В некоторых клетках стола находятся дырки, если хотя бы одна клетка слайма окажется на дырке, он упадет со стола. Слайм занимает 4 клетки, и изначально находится в верхнем левом углу поля размером  $2 \times 2$ , цель его побега — занять нижний правый угол размером  $2 \times 2$ .

Слайм может передвигаться по полю с помощью *трансформаций*. Трансформация заключается в следующем: одна из клеток слайма передвигается в другую, соседнюю с ней по стороне или углу. После трансформации слайм должен образовывать 4-связную фигуру (иными словами, между любыми двумя клетками слайма должен существовать путь по клеткам слайма, переходящий каждый раз в клетку, соседнюю по стороне).

Слайм хочет сбежать от Никиты как можно быстрее, чтобы его клетки никогда не оказывались на клетках стола с дырками. Найдите минимальное число трансформаций, за которое он сможет это сделать.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $m$  ( $2 \leq n, m \leq 3 \cdot 10^5$ ;  $n \cdot m \leq 3 \cdot 10^5$ ) — размеры стола.

В следующих  $n$  строках содержатся по  $m$  символов  $a_{i,j}$  — описание стола. Символ «#» обозначает дырку в столе, а символ «.» обозначает поверхность стола.

Гарантируется, что верхний левый и нижний правый углы стола размером  $2 \times 2$  составляют поверхность стола.

### Формат выходных данных

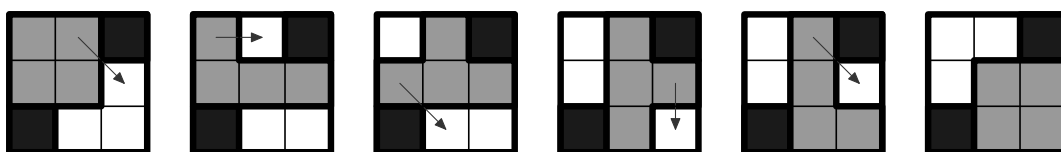
Выведите минимальное количество трансформаций, в результате которых слайм займет нижний правый угол стола размером  $2 \times 2$ , начав в верхнем левом, либо  $-1$ , если он не сможет этого сделать.

### Примеры

стандартный ввод	стандартный вывод
3 3 ..# ... #..	5
3 5 ..### ..... ##...	-1

### Замечание

На рисунке приведены трансформации для первого примера. Черным обозначены дырки в столе, серым — клетки слайма. Стрелками обозначены передвижения клеток.



## Задача К. Отходы производства

Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Крупная промышленная компания управляет заводом, который производит важные товары, а отходы экологично утилизирует. Даже в те дни, когда заводы простаивают (что может случаться часто), доля отходов все равно утилизируется безопасным образом.

Известно, что заводы работали  $n$  дней,  $i$ -й из этих дней был  $d_i$ -м производственным днем от начала работы компании, в этот день было использовано  $w_i$  сырья для производства. В остальные дни завод простаивал, будем считать, что использовалось 0 сырья.

Производство и утилизация параметризованы *коэффициентом переработки сырья*  $r$ . Чем он выше, тем быстрее происходит утилизация, но тем больше отходов производится при обработке сырья. Формально,

- Если к концу некоторого дня на производстве ровно  $x$  отходов, то в начале следующего дня их останется  $x \cdot (1 - r)$ .
- Если в начале дня на производстве ровно  $x$  отходов, то к концу дня их станет  $x + w \cdot r$ , где  $w$  — количество использованного в этот день сырья.

Иными словами, если к концу дня  $j - 1$  на производстве  $x$  отходов, а в  $j$ -й день используется  $w_j$  сырья, к концу  $j$ -го дня отходов станет  $(1 - r)x + rw_j$ . В начале самого первого дня отходов было 0.

Скоро на завод придет инспекция. Инспекцию будет интересовать информация о количестве еще не утилизированных отходов в конце каких-то дней. К сожалению, эти записи были утеряны, поэтому руководство компании просит вас помочь с ответами на вопросы инспекции, имея данные о производстве.

### Формат входных данных

В первой строке ввода через пробел даны два целых числа  $n$  и  $m$  — количество дней, в которые производились новые отходы, и количество интересующих инспекцию дней, соответственно ( $1 \leq n, m \leq 10^5$ ).

Во второй строке дано единственное вещественное число  $r$  — коэффициент переработки сырья на производстве ( $0 \leq r \leq 1$ ).

В  $i$ -й из следующих  $n$  строк через пробел даны два целых числа  $d_i$  и  $w_i$  — очередной номер производственного дня и количество использованного в этот день сырья ( $1 \leq d_i, w_i \leq 10^9$ ). Гарантируется, что все  $d_i$  различны.

В последней строке через пробел перечислены  $m$  целых чисел  $q_i$  — номера дней, интересующих инспекцию ( $1 \leq q_i \leq 10^9$ ).

### Формат выходных данных

Выведите  $m$  чисел, каждое в своей строке — количество отходов, хранящихся на фабрике в каждый из интересующих инспекцию дней.

Ваш ответ будет засчитан, если каждое из выведенных чисел имеет абсолютную или относительную ошибку не более  $10^{-6}$ .

## Примеры

стандартный ввод	стандартный вывод
3 3 0.5 2 6 3 8 4 10 1 3 5	0 5.5 3.875
5 6 0.3333333 1 27 6 27 3 27 4 27 5 27 1 2 3 4 5 6	9 6 13 17.666666 20.777777 22.851851
1 1 0 1 1000000000 1000000000	0

## Замечание

В первом примере количество отходов будет меняться следующим образом:

1. в первый день их 0;
2. во второй день их станет  $0.5 \cdot 6 = 3$ ;
3. в третий день их станет  $0.5 \cdot 3 + 0.5 \cdot 8 = 5.5$ ;
4. в четвертый день их станет  $0.5 \cdot 5.5 + 0.5 \cdot 10 = 7.75$ ;
5. в пятый день завод не производит ничего нового, и отходов становится  $0.5 \cdot 7.75 = 3.875$ .

## Задача L. Места в метро

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Василий очень любит сидеть в телефоне во время поездок в метро, и очень не любит, когда кто-то из соседей видит, чем он занимается в интернете. Вася уверен, что не единственный страдает от такой проблемы, поэтому попросил вас помочь всем любителям посидеть в одиночестве.

Рассмотрим вагон метро, в котором есть один ряд из  $n$  мест, пронумерованных от 1 до  $n$ , изначально все места свободны. Поступает  $k$  последовательных запросов. Запросы бывают двух типов:

1. « $-x$ ». Ушёл пассажир, который вошёл  $x$ -м. Гарантируется, что такой пассажир есть в вагоне.
2. « $+$ ». Пришел новый пассажир. Гарантируется, что есть хотя бы одно свободное место.

На каждый запрос второго типа следует вывести такой номер свободного места, что расстояние до ближайшего занятого — максимально. Если таких мест несколько, следует вывести место с минимальным номером. Начиная со следующего запроса это место считается занятым, до тех пор, пока его не освободят.

### Формат входных данных

В первой строке заданы два целых числа  $n$  и  $k$  — количество мест и количество запросов ( $1 \leq n \leq 10^{18}$ ;  $1 \leq k \leq 10^5$ ). В последующих  $k$  строках заданы запросы, по одному на каждой строке.

### Формат выходных данных

Для каждого запроса « $+$ » выведите номер места, куда сядет соответствующий пассажир.

### Примеры

стандартный ввод	стандартный вывод
5 9 + -1 + + + + -4 + +	1 1 5 3 2 3 4
5 8 + + + + + -4 -3 +	1 5 3 2 4 2