

## Задача А. 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 Петингого 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 В. Прямоугольники

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В связи с большим количеством покупок дачных участков, два больших, но от этого не менее гордых государства (назовем их условно «первое» и «второе»), установили ряд соглашений, касающихся участков земли около их границы.

Чтобы лучше понять нововведения, рассмотрим границу между этими государствами на карте, которая висит на стене так, что север находится сверху. Введём ортонормированную систему координат, в которой ось  $OX$  направлена с запада на восток, а  $OY$  — с юга на север. Рассмотрим  $n$  равных по величине отрезков на оси  $OX$ ,  $i$ -й из этих отрезков имеет координаты  $(i - 1, i)$ . Каждому из них сопоставим вертикальную полосу, образованную всеми возможными прямыми, параллельными  $OY$  и проходящими через сам отрезок. Теперь, чтобы разделить государства, рассмотрим придуманную систему уровней, основанную на введённых вертикальных полосах. Для каждой полосы определим  $e_j$  уровень, который задается некоторым числом  $z_i$ . Точки, принадлежащие вертикальной полосе соответствующего отрезка, лежащие выше уровня, принадлежат первому государству, а ниже — второму.

Когда коренной житель одного из государств хочет купить прямоугольный участок земли со сторонами, параллельными осям координат (участки другого вида никого не интересуют), он может это сделать, если его родное государство доминирует на выбранном участке. Это происходит, если государство доминирует на большей, чем другое государство, части вертикальных полос, образованных отрезками на оси  $OX$ . Для вертикальных полос свойство преобладания определяется следующим образом: если площадь участка на этой полосе, принадлежащего одному из государств, строго больше площади, принадлежащей другому, то первое из них доминирует на этой полосе.

Вас просят написать программу, которая могла бы определять государство, доминирующее на участке, а также изменять границу между государствами.

### Формат входных данных

В первой строке входного файла записано  $n$  — количество отрезков, на которые разделена ось  $OX$  ( $1 \leq n \leq 5 \cdot 10^4$ ).

Во второй строке —  $n$  чисел  $z_i$ , определяющих границу между государствами ( $0 \leq z_i \leq 10^9$ ).

В третьей строке задано  $m$  — число запросов к Вашей программе ( $1 \leq m \leq 10^5$ ).

Далее следует  $m$  строк с запросами. Каждый запрос имеет вид « $C x z$ » или « $Q x_1 y_1 x_2 y_2$ ». Запрос вида « $C x z$ » означает, что уровень вертикальной полосы номер  $x$  стал равным  $z$  ( $1 \leq x \leq n$ ,  $1 \leq z \leq 10^9$ ). Запрос вида « $Q x_1 y_1 x_2 y_2$ » ( $1 \leq x_1 < x_2 \leq n$ ,  $0 \leq y_1 < y_2 \leq 10^9$ ) означает, что требуется вывести государство, доминирующее на участке, левой границей которого является вертикальная полоса номер  $x_1$  (включительно), правой границей — вертикальная полоса номер  $x_2$  (включительно), а с юга и с севера участок ограничен координатами  $y_1$  и  $y_2$  соответственно.

Все числа во входном файле целые.

### Формат выходных данных

Для каждого запроса вида « $Q x_1 y_1 x_2 y_2$ » выведите «1», если на этом участке доминирует первое государство, «2», если второе, и «0», если ни у одного из государств преимущества нет.

## Пример

стандартный ввод	стандартный вывод
2	1
0 0	1
5	0
Q 1 0 2 2	
C 1 1	
Q 1 0 2 2	
C 2 1	
Q 1 0 2 2	

## Задача С. King and ICPC

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

Тёмные времена нависли над АСМ-ом. Тысячи команд АСМ-щиков вышли на защиту от нападения СТФ-щиков. Всего имеется  $N$  команд, пронумерованных от 1 до  $N$ . Каждая команда, по традиции, состоит из трёх человек. Каждому человеку присвоили параметр АСМ-овости, выраженный целым числом.

Король АСМ-щиков хочет создать армию, выбрав из каждой команды ровно по одному человеку так, чтобы суммарная АСМ-овость была как можно больше, но при этом делилась нацело на число  $D$ .

Так как АСМ-щики по своей натуре любят усложнять задачи, они решили рассмотреть несколько отрезков  $[l_i, r_i]$ , составляя армию лишь из команд с номерами из этого отрезка.

Король поручил вам обработку этих запросов. На горизонте уже видны флаги, у вас осталось меньше пяти часов!

### Формат входных данных

Первая строка содержит два целых числа  $N$  ( $1 \leq N \leq 50\,000$ ) и  $D$  ( $1 \leq D \leq 50$ ).

В следующих  $N$  строках содержится описание каждой команды в виде трёх целых чисел. АСМ-овость каждого человека является целым числом от 0 до  $10^9$ .

Следующая строка содержит целое число  $M$  — количество отрезков ( $1 \leq M \leq 300\,000$ ).

В следующих  $M$  строках идёт описание каждого запроса в виде пары чисел  $l_i$  и  $r_i$  ( $1 \leq l_i \leq r_i \leq N$ ).

### Формат выходных данных

На каждый отрезок выведите в отдельной строке максимальную сумму, которую можно на нём собрать. Если же сумму, делящуюся на  $D$  собрать невозможно, выведите «-1» (без кавычек).

### Пример

стандартный ввод	стандартный вывод
2 2	0
0 1 3	2
1 2 3	6
3	
1 1	
2 2	
1 2	

## Задача D. Простыни

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

В один прекрасный день маленький Дональд решил вымыть  $N$  своих чистых белых простыней. После мытья он положил их на землю во дворе, чтобы их высушить. Дональд помещал простыни так, чтобы никакие две простыни не касались ни сторонами, ни углами, и чтобы их стороны не пересекались, но возможно, что он разместил меньшие простыни поверх более крупных или что простыня полностью закрывает какие-то другие простыни. Сделав это, Дональд лег спать.

Друг Дональда Ким как-то узнал о том, что Дональд сушит простыни и решил пообщаться с ним. Он нашел пейнтбольный пистолет своего отца на чердаке. Наряду с пистолетом у него было несколько пейнтбольных мячей, каждый из них имел свой цвет (не обязательно уникальный). Как только Дональд заснул, Ким вошёл во двор к Дональду и начал стрелять пр простыням из пейнтбольного пистолета. Простыни Дональда очень тонкие, поэтому, когда Ким попадает в какую-то простыню, она пропускает краску дальше, на простыню ниже (и та тоже, и так происходит, пока не закончатся простыни и краска не попадет на землю). После того, как Ким использовал все шары, он с радостью покинул двор Дональда. Дональд был очень расстроен, увидев, что случилось с его простынями. Дональд очень заинтересован в правильных данных о преступлении Кима, но он в шоке и не способен думать, поэтому просит вас сказать ему количество цветов на каждой простыне.

Мы можем представлять двор Дональда как бесконечную систему координат, а простыни - как прямоугольники, параллельные осям координат. Выстрелы Кима могут быть представлены как точки в этой системе.

Когда-то в детстве дедушка рассказал Киму, что снаряд никогда не попадает в одну воронку дважды, так что координаты всех выстрелов попарно различны.

### Формат входных данных

Первая строка ввода содержит два числа - количество простынь  $N$  ( $1 \leq N \leq 300000$ ), и количество шаров  $M$  ( $1 \leq M \leq 300000$ ).

$i$ -я из следующих  $N$  строк содержит четыре числа: координаты нижнего левого угла  $A_i$ ,  $B_i$  ( $1 \leq A_i, B_i \leq 10^9$ ) и верхнего правого угла  $C_i$ ,  $D_i$ , ( $1 \leq C_i, D_i \leq 10^9$ )  $i$ -й простыни.

$j$ -я из следующих  $M$  строк содержит три числа, где  $X_j$ ,  $Y_j$  ( $1 \leq X_j, Y_j \leq 10^9$ ) - координаты  $j$ -го выстрела Кима и  $K_j$  ( $1 \leq K_j \leq 10^9$ ) - цвет  $j$ -го пейнтбольного шара.

### Формат выходных данных

$i$ -я из  $N$  выходных строк должна содержать количество цветов на  $i$ -м листе.

## Примеры

стандартный ввод	стандартный вывод
2 2 1 1 3 3 5 6 10 10 3 3 1 5 1 2	1 0
3 3 1 1 7 7 2 2 6 6 3 3 5 5 4 4 1 2 6 2 4 7 3	3 2 1
1 3 1 1 7 7 2 6 2 4 7 3 4 4 1	3

## Задача Е. Необычный массив

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 512 мегабайт

У Васи есть массив, состоящий из  $n$  чисел  $a_1, a_2, \dots, a_n$ . Для каждой позиции  $i$  и для каждого подотрезка массива  $[l, r]$ , который содержит позицию  $i$  (то есть,  $1 \leq l \leq i \leq r \leq n$ ), Вася вычисляет значение  $c_{i,l,r}$  следующим образом. Вася выписывает на листочек числа из массива с позиции  $l$  до позиции  $r$ , всего  $len = r - l + 1$  чисел (среди которых обязательно есть  $a_i$ ), и сортирует выписанные числа по возрастанию. После чего Вася находит, на какой позиции  $j$  в полученном отсортированном массиве стоит число  $a_i$ . Если таких позиций несколько, то среди них он выбирает ту, которая максимизирует расстояние от середины массива — позиции  $mid = \lceil (len + 1)/2 \rceil$  ( $len/2 + 1$  в случае четного  $len$  и  $(len + 1)/2$  в случае нечетного  $len$ ). Полученное расстояние  $|j - mid|$  и есть искомая величина  $c_{i,l,r}$ .

Например, если у Васи был массив  $a = \{5, 1, 3, 2, 1, 7\}$ , а  $i = 2, l = 2, r = 5$ , то Вася выпишет на листочек числа  $\{1, 3, 2, 1\}$ , отсортирует их и получит массив  $\{1, 1, 2, 3\}$ , длина которого равна 4. Середина этого массива находится на позиции  $4/2 + 1 = 3$ , а искомое число  $a_i = 1$  стоит в этом массиве на позициях 1 и 2. Среди этих двух позиций Вася выбирает ту, которая дальше от середины, то есть, позицию 1. Искомая разность между позициями равна 2, и это и есть значение  $c_{2,2,5}$ .

Для каждой позиции  $i$  Вася вычисляет величину  $b_i$ , которая равна максимуму среди значений  $c_{i,l,r}$  среди всех подотрезков, содержащих позицию  $i$ .

Как вы видите, определение числа  $b_i$  достаточно сложное. Помогите Васе вычислить значения  $b_i$  для всех позиций массива.

### Формат входных данных

В первой строке входных данных находится одно целое число  $n$  ( $1 \leq n \leq 200\,000$ ) — размер массива Васи.

Во второй строке находится  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq n$ ) — элементы массива.

### Формат выходных данных

В единственной строке выведите  $n$  чисел,  $i$ -е из них должно быть равно  $b_i$ .

### Примеры

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

### Замечание

Разберем подробнее первый пример.

- Для первой позиции Вася рассмотрит все подотрезки, содержащие эту позицию, в частности, подотрезок  $[1, 5]$ , где  $l = 1$  и  $r = 5$ . Для вычисления  $c_{1,l,r} = c_{1,1,5}$  Вася выпишет числа  $\{5, 4, 3, 2, 1\}$  и после сортировки получит  $\{1, 2, 3, 4, 5\}$ . Середина этого массива находится на позиции 3, а искомое число  $a_1 = 5$  — на позиции 5. Таким образом,  $c_{1,1,5} = 2$ . Нетрудно заметить, что это число — максимальное среди всех подотрезков, содержащих позицию 1, а значит,  $b_1 = 2$ .
- $b_2 = c_{2,2,4}$ .
- $b_3 = c_{3,3,5}$ .

- $b_4 = c_{4,1,4}$ . Действительно, если выписать числа на подотрезке  $[1, 4]$ , то получится массив  $\{5, 4, 3, 2\}$ , который после сортировки превратится в  $\{2, 3, 4, 5\}$ . Середина этого массива находится на позиции 3, а искомый элемент  $a_4 = 2$  — на позиции 1. Таким образом,  $c_{4,1,4} = 2$ .
- $b_5 = c_{5,1,5}$ .



## Задача F. Размен денег

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 256 мегабайт

За свою долгую жизнь Боря собрал коллекцию из  $n$  монет. Он выложил все эти монеты в ряд. При этом  $i$ -я в ряду монета имеет номинал  $a_i$ .

Боря собирается в очередное путешествие, но у него осталось очень мало времени на сборы. Поэтому он хочет взять некоторый отрезок лежащих подряд монет и надеется, что ему их хватит.

Боря хочет ответить на несколько запросов. В каждом запросе Боря хочет узнать, какую минимальную сумму он не сможет заплатить без сдачи, если он возьмет все монеты с  $l_i$ -й по  $r_i$ -ю. Более формально, он хочет найти такое минимальное натуральное число  $z$ , что нельзя выбрать подмножество монет с номерами от  $l_i$  до  $r_i$ , суммарный номинал которых равен  $z$ .

### Формат входных данных

В первой строке задано два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 150\,000$ ) — количество монет у Бори и количество запросов. В следующей строке задано  $n$  чисел  $a_i$  ( $1 \leq a_i \leq 10^9$ ) — номинал  $i$ -й монеты.

В следующих  $m$  строках задано по два числа  $l_i$  и  $r_i$  ( $1 \leq l_i \leq r_i \leq n$ ) — описание запросов.

### Формат выходных данных

На каждый из  $m$  запросов выведите минимальную сумму, которую нельзя заплатить без сдачи, воспользовавшись монетами с  $l_i$ -й по  $r_i$ -ю.

### Пример

стандартный ввод	стандартный вывод
5 5	13
2 1 5 3 1	4
1 5	1
1 3	2
1 1	11
2 4	
2 5	

## Задача G. Шахматные баталии

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Ильдар и Ваня устали постоянно играть в шахматы, поэтому они придумали новую шахматную игру.

Игра происходит на шахматном поле размером  $2n \times 2m$ . Это поле имеет  $2n$  строк и  $2m$  столбцов. Клетки этого поля покрашены в черный и белый цвета шахматной раскраской. Более точно, клетка в  $i$ -й строке и  $j$ -м столбце имеет белый цвет, если  $i + j$  чётно, и чёрный цвет в противном случае.

Игра устроена следующим образом. Ильдар помечает некоторые **белые** клетки поля как недоступные. После этого он предлагает Ване попробовать решить следующую задачу: может ли он на доступных **белых** клетках поля расставить  $n \times m$  шахматных королей так, что никакие два выставленных короля не бьют друг друга, то есть не стоят на клетках поля, соседних по стороне или углу.

Конечно, Ильдар планирует сделать игру интересной и предложить Ване несколько сложных комбинаций недоступных клеток. Для этого он попросил у вас помощи. Чтобы перед игрой понять, как лучше всего действовать, он хочет потренироваться. Для этого, он берёт пустое поле и хочет  $q$  раз либо сделать какую-то доступную клетку недоступной, либо сделать какую-то недоступную клетку доступной. После каждого изменения он бы хотел знать, каким будет ответ на задачу для Вани для текущего множества недоступных клеток.

Помогите Ильдару сделать игру интересной! Напишите программу, которая будет отвечать на его запросы.

### Формат входных данных

В первой строке находится три целых числа  $n, m, q$  ( $1 \leq n, m, q \leq 200\,000$ ) — количество пар строк шахматной доски, количество пар столбцов шахматной доски и количество запросов.

Следующие  $q$  строк описывают запросы Ильдара. Каждая из этих строк содержит два целых числа  $i, j$  ( $1 \leq i \leq 2n, 1 \leq j \leq 2m, i + j$  чётно). Если клетка  $(i, j)$  была недоступна, она становится доступной, в противном случае она становится недоступной.

### Формат выходных данных

Выведите  $q$  строк. В  $i$ -й из этих строк выведите ответ на задачу для доски, полученной после  $i$  первых запросов Ильдара.

Выведите «YES», если Ваня может так расставить шахматных королей на незанятые белые клетки поля, что никакие два короля не будут бить друг друга, и «NO», иначе.

## Примеры

стандартный ввод	стандартный вывод
1 3 3	YES
1 1	YES
1 5	NO
2 4	
3 2 10	YES
4 2	YES
6 4	NO
1 3	NO
4 2	YES
6 4	YES
2 2	NO
2 4	YES
1 3	YES
4 4	NO
3 1	

## Замечание

В первом примере, после второго запроса пешки будут стоять на клетках (1, 1) и (1, 5). Тогда Ваня может поставить три короля на клетки (2, 2), (2, 4) и (2, 6).

После третьего запроса пешки будут стоять на клетках (1, 1), (1, 5) и (2, 4). Тогда отстает всего три пустые клетки (2, 2), (1, 3) и (2, 6). Ваня не может поставить трех королей на эти клетки, потому что короли в клетках (2, 2) и (1, 3) бьют друг друга, так как эти клетки соседние по углу.

## Задача Н. Нуль

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Вам дана последовательность  $a = a_1, a_2, \dots, a_N$ , состоящая из 0 и 1 и последовательность  $b = b_1, b_2, \dots, b_N$ , состоящая из 0 и 1. Длина каждой из последовательностей равняется  $N$ .

Вы можете выполнять  $Q$  видов операций, где  $i$ -я операция имеет следующий вид:

- Замените значения  $a_{l_i}, a_{l_i+1}, \dots, a_{r_i}$  на 1.

Выполните какие-то из данных операций так, чтобы минимизировать количество индексов  $i$  таких, что  $a_i \neq b_i$ .

### Формат входных данных

В первой строке входных данных содержится целое число  $N$  ( $1 \leq N \leq 200,000$ ) — количество элементов в последовательностях. Во второй строке содержатся  $N$  целых чисел  $b_i$  — элементы последовательности  $b$ , равные 0 или 1. В третьей строке содержится целое число  $Q$  ( $1 \leq Q \leq 200,000$ ) — количество возможных операций. В следующих  $Q$  строках содержатся пары целых чисел  $l_i, r_i$  ( $1 \leq l_i \leq r_i \leq N$ ) — возможные операции.

### Формат выходных данных

В единственной строке выведите минимальное количество индексов  $i$  таких, что  $a_i \neq b_i$  после выполнения какого-то множества операций.

## Примеры

стандартный ввод	стандартный вывод
3 1 0 1 1 1 3	1
3 1 0 1 2 1 1 3 3	0
3 1 0 1 2 1 1 2 3	1
5 0 1 0 1 0 1 1 5	2
9 0 1 0 1 1 1 0 1 0 3 1 4 5 8 6 7	3
15 1 1 0 0 0 0 0 0 1 0 1 1 1 0 0 9 4 10 13 14 1 7 4 14 9 11 2 6 7 8 3 12 7 13	5
10 0 0 0 1 0 0 1 1 1 0 7 1 4 2 5 1 3 6 7 9 9 1 5 7 9	1

## Задача I. Сложная задача

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	768 мегабайт

У Егора есть большой лист клетчатой бумаги и не менее большое домашнее задание по математике. Егор, как недобросовестный ученик, планирует списать домашнюю работу с сервиса Титан Бета.

Решение задачи Егора в системе Титан Бета состоит из  $n$  формул-шагов, каждую из которых он должен перенести на лист. Разумеется, формулы в листе должны идти по порядку — слева направо. Если в какой-то момент написать следующую формулу нельзя — её обязательно надо перенести на следующую строку. Но также разрешено делать переносы и в произвольном месте решения, на усмотрение ученика, однако формула должна быть написана целиком на одной строке.

Поскольку в формулах могут возникать дробные выражения, высоты формул бывают разными. Егор относится к формулам, как к бессвязному набору символов, который влезает в какой-то клетчатый прямоугольник; а именно,  $i$ -я формула занимает прямоугольник  $w_i$  клеток в ширину и  $h_i$  клеток в высоту.

Чтобы строки читались аккуратно, Егор выравнивает все формулы, записанные в одной строке, по нижней границе строки. Таким образом, высота строки равна высоте самой высокой формуле строки. Ширина же строки равна суммарной ширине формул в строке. Суммарной высотой решения назовем суммарную высоту всех строк.

Ширина листа бумаги составляет ровно  $m$  клеток. Егора заинтересовало, какую наименьшую высоту может иметь его решение? Поскольку ему надо делать домашнюю работу, то ответ на этот вопрос предстоит найти вам.

### Формат входных данных

В данной задаче ввод устроен необычным образом — для тестов последней группы часть входного файла вам придется генерировать самостоятельно.

В первой строке вводятся три числа  $n, m, k$  ( $1 \leq n \leq 10^7$ ,  $1 \leq m \leq 10^{11}$ ,  $1 \leq k \leq \min(500\,000, n)$ ) — общее количество формул, ширина листа, количество формул, высота и ширина для которых будут введены далее.

В следующих  $k$  строках вводится по три целых числа  $i_j, w_{i_j}, h_{i_j}$  ( $1 \leq i_j \leq n$ ,  $1 \leq w_{i_j}, h_{i_j} \leq m$ ). Первое число — номер одной из формул Егора. Следующие два числа — ширина и высота этой формулы соответственно.

Гарантируется, что индексы  $i_j$  возрастают с каждой следующей строкой.

В последней строке вводятся 4 числа  $a, b, c, d$  ( $1 \leq a, c \leq \min(10\,000, m)$ ,  $0 \leq b, d \leq m$ ) — вспомогательные параметры, которые отвечают за генерацию остальных входных данных.

Для тех формул, номера которых не были перечислены во входных данных, высоту и ширину придётся генерировать самостоятельно. А именно, для любого  $i$ , не перечисленного во входных данных,  $w_i = ((w_{i-1} \cdot a + b) \bmod m) + 1$ ,  $h_i = ((h_{i-1} \cdot c + d) \bmod m) + 1$ . Гарантируется, что  $i = 1$  присутствует во входных данных (соответственно, размеры всех формул определяются однозначно).

### Формат выходных данных

Выведите одно число — минимальную высоту листа шириной  $m$ , в который получится уместить все формулы.

## Примеры

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

## Замечание

В первом примере из условия оптимальным вариантом размещения будет тот, при котором первая формула находится на отдельной строке, а вторая и третья — на следующей. Таким образом, суммарная высота получается равной  $2 + 3 = 5$  клеткам.

Во втором примере из условия прямоугольники имеют вид  $[(2, 3), (1, 2), (3, 1), (2, 3)]$ . Третий прямоугольник обязательно займет целую строку. Четвертая формула, таким образом, займет свою отдельную строчку, а первая и вторая формулы сгруппируются. Получившиеся строки будут вместе занимать  $3 + 1 + 3 = 7$  клеток в высоту.

## Система оценки

Тесты к этой задаче состоят из восьми групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **необходимых** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Дополнительные ограничения	Необх. группы	Комментарий
		$n$		
0	0	—	—	Тесты из условия.
1	12	$n \leq 10, n = k$	—	
2	13	$n \leq 100, n = k$	1	
3	14	$n \leq 3000, n = k$	1, 2	
4	7	$n \leq 500\,000, n = k$	—	$h_i < h_{i+1}$
5	8	$n \leq 500\,000, n = k$	—	$h_i > h_{i+1}$
6	22	$n \leq 500\,000, n = k$	1–5	
7	22	$n \leq 5 \cdot 10^6$	0–6	
8	2	—	0–7	<b>Offline-проверка.</b>

## Задача J. Фокус с подмножествами

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Ваня придумал интересный фокус с множеством целых чисел.

Пусть у фокусника есть множество положительных целых чисел  $S$ . Он называет некоторое положительное целое число  $x$ . Зритель должен выбрать, не показывая фокуснику, некоторое подмножество  $S$  (возможно пустое). После этого зритель называет фокуснику размер выбранного подмножества. Фокус заключается в том, что после этого фокусник отгадывает: верно ли, что сумма элементов выбранного подмножества не превосходит  $x$ . Для пустого подмножества сумма предполагается равной 0.

Ване очень понравился этот фокус, поэтому он начал готовиться к тому, чтобы показать его публике. Для этого он приготовил некоторое множество **различных** положительных целых чисел  $S$ . Конечно, Ваня хочет, чтобы фокус обязательно получился. Он называет положительное целое число  $x$  *неудачным*, если не может быть точно уверен, что фокус пройдет удачно для любого подмножества, которое выберет зритель.

Чтобы оценить, насколько хорошее множество  $S$  он выбрал, он хочет посчитать количество неудачных положительных целых чисел для него.

Также Ваня планирует протестировать различные множества  $S$ . Поэтому он просит вас написать программу, которая найдет количество неудачных положительных целых чисел для изначального множества  $S$  и для множества  $S$  после каждого изменения. Ваня сделает  $q$  изменений своего множества, каждое изменение будет одного из двух видов:

- Добавить новое число  $a$  в множество  $S$ .
- Удалить некоторое число  $a$  из множества  $S$ .

### Формат входных данных

В первой строке находятся два целых числа  $n, q$  ( $1 \leq n, q \leq 200\,000$ ) — размер изначального множества  $S$  и количество изменений.

В следующей строке находятся  $n$  **различных** целых чисел  $s_1, s_2, \dots, s_n$  ( $1 \leq s_i \leq 10^{13}$ ) — элементы изначального множества  $S$ .

В каждой из следующих  $q$  строк находятся два целых числа  $t_i, a_i$  ( $1 \leq t_i \leq 2, 1 \leq a_i \leq 10^{13}$ ), описывающих очередное изменение.

- Если  $t_i = 1$ , то это операция добавления нового числа  $a_i$  в множество  $S$ . Гарантируется, что этого числа не было в множестве  $S$  до выполнения операции.
- Если  $t_i = 2$ , то это операция удаления числа  $a_i$  из множества  $S$ . Гарантируется, что это число было в множестве  $S$  до выполнения операции.

### Формат выходных данных

Выведите  $q + 1$  строку.

В первой строке выведите количество неудачных положительных целых чисел для изначального множества  $S$ . В следующих  $q$  строках выведите количество неудачных положительных чисел для множества  $S$  после каждого изменения.



## Пример

стандартный ввод	стандартный вывод
3 11	4
1 2 3	1
2 1	6
1 5	12
1 6	19
1 7	13
2 6	8
2 2	2
2 3	10
1 10	3
2 5	0
2 7	0
2 10	

## Замечание

В первом тесте изначальное  $S = \{1, 2, 3\}$ . Для этого множества фокус может не получиться при  $x \in \{1, 2, 3, 4\}$ . Например, если  $x = 4$ , то зритель может загадать подмножество  $\{1, 2\}$ , сумма элементов которого равна  $3 \leq x$ , а может загадать подмножество  $\{2, 3\}$ , сумма элементов которого равна  $5 > x$ . Однако в обоих случаях зритель назовет фокуснику размер подмножества 2, поэтому он не сможет точно сделать правильный ответ. При этом поскольку подмножество размера 3 единственно, а сумма в любом подмножестве меньшего размера не превосходит 5, все  $x \geq 5$  не являются неудачными.

## Система оценки

Тесты к этой задаче состоят из семи групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов всех необходимых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Ограничения			Необх. группы	Комментарий
		$n$	$q$	Доп. ограничения		
0	0	—	—	—	—	Тесты из условия.
1	10	$n \leq 12$	$q \leq 100$	все $ S  \leq 12$	0	
2	10	$n \leq 100$	$q \leq 100$	$s_i, a \leq 100$	0	
3	20	$n \leq 2000$	$q \leq 2000$	—	0 – 2	
4	15	$n \leq 50\,000$	$q \leq 50\,000$	—	0 – 3	
5	15	$n \leq 100\,000$	$q \leq 100\,000$	—	0 – 4	
6	15	$n \leq 150\,000$	$q \leq 150\,000$	—	0 – 5	<b>Offline-проверка.</b>
7	15	—	—	—	0 – 6	<b>Offline-проверка.</b>

## Задача К. База данных

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Заведующая учебной частью Байтландской школы в целях контроля успеваемости издала приказ, согласно которому каждый школьник обязан в конце каждой недели предоставлять информацию о своих оценках. Требуется предоставлять лишь свой средний балл, который по мнению завуча полностью характеризует успешность школьника в течение недели.

Родителям понравилось это нововведение, ведь теперь они могут посмотреть уровень успеваемости их ребёнка относительно других школьников. Каждую субботу ученики посылают в базу данных школы свои средние оценки, которые там сохраняются, а затем, родители учеников выполняют  $m$  запросов к данной базе данных.

Пусть  $u$  — максимальное число, находящееся в базе в данный момент времени, а  $cnt(x)$  — количество чисел больше либо равных  $x$  (с повторениями), находящихся в базе. Предусмотрены четыре вида запросов к базе данных:

1. Заменить хранимые в базе числа на последовательность  $(cnt(1), cnt(2), \dots, cnt(u))$ .
2. Добавить в базу некоторое число  $x$ .
3. Удалить из базы одно вхождение некоторого числа  $x$ , если такое имеется.
4. Вывести количество чисел, равных данному числу  $x$ .

Родители начинают запрашивать интересующую их информацию и модернизировать данные лишь после того, как все  $n$  школьников отправят свои оценки.

К сожалению, недавно у школы кончилась лицензия на эту замечательную базу данных, поэтому пока её не продлили, на все запросы придётся отвечать вам.

### Формат входных данных

Первая строка входных данных содержит два числа  $n$  и  $m$  ( $1 \leq n, m \leq 200\,000$ ) — количество школьников и количество запросов к базе данных соответственно.

Во второй строке записаны  $n$  целых чисел  $g_i$  ( $1 \leq g_i \leq 200\,000$ ) — значения средних оценок школьников.

Далее следуют  $m$  строк, описывающих запросы в порядке их применения. Каждое описание начинается с одной из букв «t», «a», «r» или «c», означающих запрос первого, второго, третьего или четвертого типа соответственно. Для запросов второго, третьего и четвертого типа далее следует число  $x_i$  ( $1 \leq x_i \leq 200\,000$ ) — параметр запроса.

### Формат выходных данных

Сначала выведите ответы на все запросы четвертого типа. Затем выведите все числа, находящиеся в базе после выполнения всех запросов, в порядке неубывания.

Гарантируется, что в выходных данных должно будет присутствовать хотя бы одно число.

## Пример

стандартный ввод	стандартный вывод
6 8	0
4 3 3 3 6 6	2
t	3 3 5 7 7
c 4	
a 5	
a 3	
r 5	
c 2	
t	
r 3	

## Замечание

Рассмотрим как меняется последовательность из примера:

1. (4, 3, 3, 3, 6, 6)
2. (6, 6, 6, 3, 2, 2)
3. (6, 6, 6, 3, 2, 2, 5)
4. (6, 6, 6, 3, 2, 2, 5, 3)
5. (6, 6, 6, 3, 2, 2, 3)
6. (7, 7, 5, 3, 3, 3)
7. (7, 7, 5, 3, 3)

Ниже предоставлены критерии оценки:

№	Баллы	Ограничения	Необх. группы	Комментарий
0	0	—	—	Тесты из условия.
1	15	$n, m, x_i \leq 200$	0	—
2	15	$n, m, x_i \leq 3000$	0 – 1	—
3	35	—	—	Запросы только 1 и 4 типа.
4	35	—	0 – 3	—

## Задача L. Эффективное тестирование

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Начиная с 20xx года все организаторы всех школьных олимпиад по программированию договорились проводить соревнования исключительно по интернету, для чего было создано общество с ограниченной ответственностью «Организация онлайн-олимпиад» (ООО «ООО»). Разумеется, такая серьёзная организация не может обойтись без собственной тестирующей системы, поэтому для её создания были наняты эффективные менеджеры, закуплены доски и подготовлена синяя изолента.

Для повышения эффективности процесса тестирования была разработана следующая архитектура. Сначала все  $m$  тестов задачи располагаются в порядке от 1 к  $m$  в очереди тестирования. Затем модуль планирования последовательно выполняет  $n$  действий. Действие  $i$  состоит в том, чтобы выбрать отрезок очереди с позиции  $l_i$  по  $r_i$  включительно (в нумерации с единицы) и проверить решение на каждом втором тесте на этом отрезке, а именно на тестах на позициях  $l_i, l_i+2, l_i+4, \dots, r_i$  очереди (при этом гарантируется, что  $l_i$  и  $r_i$  имеют одинаковую чётность). После этого те тесты, на которых было проведено тестирование, удаляются из очереди, а все оставшиеся тесты сдвигаются по очереди таким образом, чтобы пустых мест не осталось. Например, если в очереди находились тесты с исходными номерами 2, 3, 4, 5, 10, 12, 13, 20 и была применена операция с  $l_i = 3, r_i = 7$ , то посылка будет протестирована на тестах с позиций 3, 5 и 7, которые исходно имели номера 4, 10 и 13. После выполнения данной операции очередь тестирования будет состоять из тестов с исходными номерами 2, 3, 5, 12, 20.

Вам поручено реализовать модуль, который для каждого из  $n$  описанных выше действий будет определять минимальный и максимальный номер теста в изначальной нумерации из тех, на которых на этом шаге проверялось решение.

### Формат входных данных

В первой строке входных данных находятся два числа  $n$  и  $m$  ( $1 \leq n \leq 100\,000, 1 \leq m \leq 10^{18}$ ) — количество действий модуля планирования и количество тестов в задаче.

В каждой из последующих  $n$  строк записаны два целых числа  $l_i$  и  $r_i$  ( $1 \leq l_i \leq r_i \leq m$ ) — параметры  $i$ -го действия модуля планирования. Гарантируется, что перед началом выполнения действия  $i$  в очереди тестирования находятся хотя бы  $r_i$  тестов и что числа  $l_i$  и  $r_i$  имеют одинаковую чётность.

### Формат выходных данных

Для каждого из  $n$  действий модуля планирования выведите два целых числа — минимальный и максимальный номер теста в исходной нумерации из тех, на которых проверялось решение на соответствующем шаге.

### Примеры

стандартный ввод	стандартный вывод
2 10	2 8
2 8	1 5
1 3	
4 6	1 1
1 1	2 2
1 1	3 3
1 1	5 5
2 2	

### Замечание

Рассмотрим, как изменяется очередь тестирования в первом примере.

- Изначально в очереди тестирования находятся все тесты от 1 до 10, то есть очередь имеет вид 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

2. При выполнении первого запроса будут удалены тесты 2, 4, 6, 8, и очередь примет вид 1, 3, 5, 7, 9, 10.
3. При выполнении второго запроса будут удалены тесты 1 и 5, очередь примет вид 3, 7, 9, 10.

## Задача М. Минимизируй количество инверсий

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

Дана перестановка  $p$  длины  $n$ .

Можно выбрать любую её подпоследовательность, удалить её из перестановки и приписать в начало перестановки в том же порядке.

Для каждого  $k$  от 0 до  $n$  найдите минимальное количество инверсий в перестановке, которое можно получить, выбрав подпоследовательность длины ровно  $k$ .

### Формат входных данных

В первой строке задано одно целое число  $t$  ( $1 \leq t \leq 50\,000$ ) — количество наборов входных данных.

В первой строке каждого набора вводится число  $n$  ( $1 \leq n \leq 5 \cdot 10^5$ ) — длина перестановки.

Во второй строке каждого набора вводится перестановка  $p_1, p_2, \dots, p_n$  ( $1 \leq p_i \leq n$ ).

Гарантируется, что сумма  $n$  не превосходит  $5 \cdot 10^5$ .

### Формат выходных данных

Для каждого набора выведите  $n + 1$  число, где  $i$ -е число — ответ на задачу для длины подпоследовательности, равной  $i - 1$ .

### Пример

стандартный ввод	стандартный вывод
3	0 0
1	4 2 2 1 4
1	5 4 2 2 1 5
4	
4 2 1 3	
5	
5 1 3 2 4	

### Замечание

Во втором наборе:

- Для длины 0:  $[4, 2, 1, 3] \rightarrow [4, 2, 1, 3]$ : 4 инверсии.
- Для длины 1:  $[4, 2, 1, 3] \rightarrow [1, 4, 2, 3]$ : 2 инверсии.
- Для длины 2:  $[4, 2, 1, 3] \rightarrow [2, 1, 4, 3]$  или  $[4, 2, 1, 3] \rightarrow [1, 3, 4, 2]$ : 2 инверсии.
- Для длины 3:  $[4, 2, 1, 3] \rightarrow [2, 1, 3, 4]$ : 1 инверсия.
- Для длины 4:  $[4, 2, 1, 3] \rightarrow [4, 2, 1, 3]$ : 4 инверсии.

## Задача N. 3D сумма

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 8 секунд  
Ограничение по памяти: 512 мегабайт

Задан трехмерный массив  $a$  размера  $n_x \times n_y \times n_z$  ( $n_x n_y n_z \leq 10^6$ ). Изначально он заполнен 0. Массив индексируется с 0.

Ответьте на два вида запросов в online:

- а  $x_1 y_1 z_1 x_2 y_2 z_2 v$  — прибавить ко всем элементам  $a_{x,y,z}$  таким, что  $x_1 \leq x < x_2$ ,  $y_1 \leq y < y_2$  и  $z_1 \leq z < z_2$ ;
- с  $x_1 y_1 z_1 x_2 y_2 z_2$  — вычислить сумму всех  $a_{x,y,z}$  таких, что  $x_1 \leq x < x_2$ ,  $y_1 \leq y < y_2$  и  $z_1 \leq z < z_2$ .

Все вычисления проводить по модулю  $2^{32}$ .

### Формат входных данных

В первой строке заданы три целых числа  $n_x, n_y$  и  $n_z$  ( $1 \leq n_x, n_y, n_z \leq 10^6$ ;  $n_x n_y n_z \leq 10^6$ ).

Во второй строке задано целое число  $q$  — число запросов ( $1 \leq q \leq 10^5$ ).

В следующих  $q$  строках заданы запросы, в каждом запросе  $0 \leq x_1 < x_2 \leq n_x$ ,  $0 \leq y_1 < y_2 \leq n_y$ ,  $0 \leq z_1 < z_2 \leq n_z$ ,  $0 \leq v \leq 10^9$ .

### Формат выходных данных

Выведите ответ: по одному целому числу на каждый запрос.

### Система оценки

Подзадача 1 (60 баллов):  $n_y = n_z = 1$

Подзадача 2 (30 баллов):  $n_z = 1$

Подзадача 3 (10 баллов): нет дополнительных ограничений

### Пример

стандартный ввод	стандартный вывод
3 2 4	0
5	20
a 0 0 0 1 2 3 5	42
s 1 1 1 2 2 2	
a 1 1 2 3 2 4 3	
s 0 0 0 2 2 2	
s 0 0 0 3 2 4	

## Задача О. Прибавление

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Дан двумерный массив целых чисел  $a$ , состоящий из  $n$  строк и  $m$  столбцов. Изначально все числа в массиве равны нулю. Необходимо обработать  $q$  запросов следующего вида:

- 1  $r_1$   $c_1$   $r_2$   $c_2$   $t$  ( $1 \leq r_1 \leq r_2 \leq n$ ,  $1 \leq c_1 \leq c_2 \leq m - t + 1$ ,  $1 \leq t \leq m$ ). Назовём операцией *прибавления в прямоугольнике*, заданном углами  $(x_1, y_1)$  и  $(x_2, y_2)$ , где  $x_1 \leq x_2$ ,  $y_1 \leq y_2$ , процедуру, в результате которой ко всем элементам массива  $a_{x,y}$  для которых  $x_1 \leq x \leq x_2$ ,  $y_1 \leq y \leq y_2$ , прибавляется 1.

Необходимо для всех  $i$ , таких что  $0 \leq i < t$  выполнить операцию прибавления в прямоугольнике, заданном углами  $(r_1, c_1 + i)$ ,  $(r_2, c_2 + i)$ .

- 2  $r_1$   $c_1$   $r_2$   $c_2$  ( $1 \leq r_1 \leq r_2 \leq n$ ,  $1 \leq c_1 \leq c_2 \leq m$ ). Необходимо вычислить сумму элементов  $a_{x,y}$  в прямоугольнике, заданном углами  $(r_1, c_1)$ ,  $(r_2, c_2)$ . Поскольку данная сумма может быть очень большой, надо вычислить остаток от её деления на  $2^{31}$ .

### Формат входных данных

В первой строке заданы три целых числа  $n$ ,  $m$  и  $q$  ( $1 \leq n, m \leq 10^9$ ,  $1 \leq q \leq 100\,000$ ) — размеры массива и количество запросов.

В следующих  $q$  строках находятся описания запросов в формате, описанном выше.

### Формат выходных данных

Для каждого запроса второго типа выведите ответ в отдельной строке. Гарантируется, что хотя бы один запрос имеет тип 2.



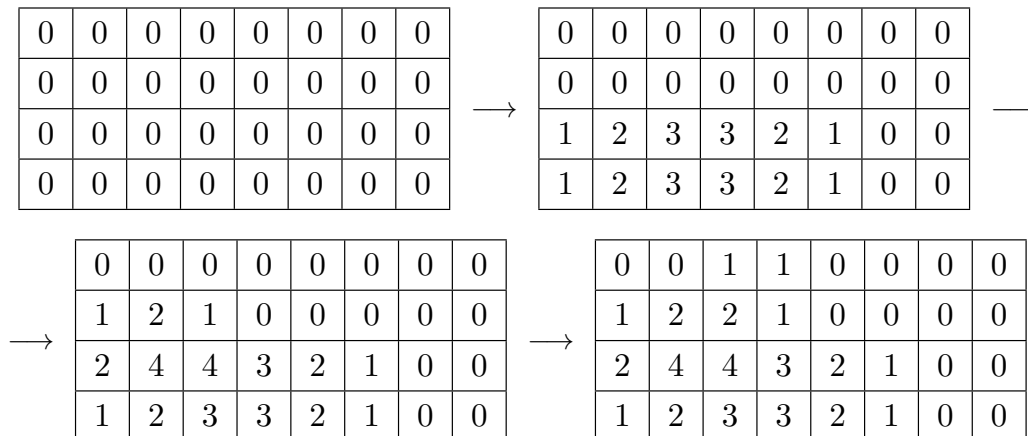
## Примеры

стандартный ввод	стандартный вывод
1 4 2 1 1 1 1 2 3 2 1 1 1 4	6
4 8 6 1 1 1 2 4 3 1 2 1 3 2 2 1 3 3 4 4 1 2 3 3 4 6 2 1 2 3 5 2 1 5 4 8	5 28 6
6 6 6 1 1 1 3 5 1 1 2 1 4 4 3 2 1 1 6 5 1 2 2 6 5 1 1 2 2 5 6 1 2 2 2 3 4	48 34
13 8 8 2 2 1 8 5 1 4 2 7 8 1 2 1 2 8 4 1 3 2 13 5 2 1 4 2 8 5 3 1 2 2 9 4 1 1 4 2 9 5 2 2 3 2 7 5	0 12 130

## Замечание

В первом примере мы имеем дело с одномерным массивом длины 4. Изначально массив равен  $(0, 0, 0, 0)$ . После первого запроса к элементам на отрезках  $[1; 2]$ ,  $[2; 3]$  и  $[3; 4]$  прибавляем по 1, и получается массив  $(1, 2, 2, 1)$  с суммой элементов 6.

Во втором примере в результате первых трёх запросов происходят следующие преобразования (строки нумеруются снизу вверх):



## Система оценки

Тесты к этой задаче состоят из девяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **необходимых** групп.

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		$n, m$	$q$		
0	0	–	–	–	Тесты из условия.
1	7	$n, m \leq 100$	$q \leq 100$	0	
2	8	$n, m \leq 700$	$q \leq 700$	0 – 1	
3	10	–	$q \leq 10\,000$	–	$t_i = 1$ .
4	15	–	$q \leq 10\,000$	0 – 3	
5	9	–	–	–	В запросах второго типа $(r_{i,1}, c_{i,1}) = (1, 1)$ , $(r_{i,2}, c_{i,2}) = (n, m)$ .
6	11	$n, m \leq 1000$	–	–	Все запросы типа 2 идут после запросов типа 1, $t_i = 1$ .
7	10	$n, m \leq 1000$	–	6	Все запросы типа 2 идут после запросов типа 1.
8	14	–	–	3, 6	$t_i = 1$ .
9	16	–	–	0 – 8	

## Задача Р. Дима и массив

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 6.5 секунд  
Ограничение по памяти: 768 мегабайт

Диме не дарили массив  $a$ , состоящий из  $n$  целых чисел на день рождения, он не покупал его, не находил на улице, а он у него просто есть и всегда был, и Диме не очень-то и интересно откуда.

Дима не играет с массивом, не дарит его Пете, не режет на кусочки и не стремится его уничтожить. Дима просто выполняет операции двух видов со своим массивом:

- $? \ l \ r$  — узнать МЕХ мультимножества  $\{a_l, a_{l+1}, \dots, a_r\}$
- $! \ i \ x$  — присвоить  $a_i$  значение  $x$  ( $0 \leq x \leq n$ )

МЕХ мультимножества чисел  $\{a_1, a_2, \dots, a_k\}$  — это минимальное целое  $t \geq 0$  такое, что  $t \neq a_i$  для всех  $1 \leq i \leq k$ .

На самом деле, Диме не очень нравится выполнять операции двух видов со своим массивом. Диму волнуют лишь результаты операций первого типа. Помогите Диме и напишите программу, которая выполнит операции за него.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $q$  ( $1 \leq n \leq 500\,000, 1 \leq q \leq 250\,000$ ) — размер массива, который есть у Димы и количество операций, соответственно.

Вторая строка содержит  $n$  целых чисел  $a_i$  ( $0 \leq a_i \leq n$ ) — массив Димы до начала операций.

Каждая из следующих  $q$  строк содержит описание одной операции в формате, описанном выше.

Гарантируется, что суммарно Дима сделал **не более 50 000 операций изменения массива**.

Элементы массива пронумерованы, начиная с 1.

### Формат выходных данных

Для каждой операция первого типа выведите одно целое число — МЕХ соответствующего мультимножества. Ответы на запросы выводите в порядке, в котором они заданы во входных данных.

### 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.

## Система оценки

Тесты к этой задаче состоят из тринадцати подзадач. Баллы за каждую подзадачу ставятся только при прохождении всех тестов подзадачи и всех тестов **необходимых** подзадач. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Доп. ограничения			Необх. группы	Комментарий
		$n$	$q$	$a_i, x$		
0	0	—	—	—	—	Тесты из условия.
1	11	$n \leq 100$	$q \leq 100$	—	0	
2	8	$n \leq 5000$	$q \leq 5000$	—	0, 1	
3	12	—	—	$a_i, x \leq 10$	0	
4	19	—	—	—	—	Нет операций изменения.
5	5	$n \leq 100\,000$	$q \leq 100\,000$	—	0 – 2	
6	5	$n \leq 150\,000$	$q \leq 150\,000$	—	0 – 2, 5	
7	5	$n \leq 200\,000$	$q \leq 200\,000$	—	0 – 2, 5, 6	
8	5	$n \leq 250\,000$	—	—	0 – 2, 5 – 7	
9	6	$n \leq 300\,000$	—	—	0 – 2, 5 – 8	<b>Offline-проверка.</b>
10	6	$n \leq 350\,000$	—	—	0 – 2, 5 – 9	<b>Offline-проверка.</b>
11	6	$n \leq 400\,000$	—	—	0 – 2, 5 – 10	<b>Offline-проверка.</b>
12	6	$n \leq 450\,000$	—	—	0 – 2, 5 – 11	<b>Offline-проверка.</b>
13	6	—	—	—	0 – 12	<b>Offline-проверка.</b>

## Задача Q. Разбиение

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

Дан массив  $a$  длины  $n$  и  $q$  запросов. В  $i$ -м запросе вводятся два целых числа  $l_i$  и  $r_i$ . Требуется разбить отрезок  $[l_i, r_i]$  массива  $a$  на три непересекающихся отрезка так, чтобы сумма максимальных элементов в каждом из отрезков была как можно меньше.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $q$  ( $3 \leq n \leq 250\,000$ ,  $1 \leq q \leq 250\,000$ ) — размер массива и количество запросов.

Вторая строка содержит  $n$  целых чисел  $a_1, \dots, a_n$  ( $1 \leq a_i \leq 10^8$ ) — элементы массива.

Далее следует  $q$  строк,  $i$ -я из которых содержит два целых числа  $l_i$  и  $r_i$  ( $1 \leq l_i \leq r_i \leq n$ ,  $r_i - l_i \geq 2$ ) — описание  $i$ -го запроса.

### Формат выходных данных

Для каждого запроса выведите минимальную сумму максимальных элементов среди всех возможных разбиений отрезка запроса на три подотрезка.

### Примеры

стандартный ввод	стандартный вывод
7 5 4 3 1 1 4 5 2 1 7 2 4 3 5 1 5 4 7	10 5 6 9 8
10 15 8 3 8 10 1 5 3 1 6 4 4 6 2 5 6 9 8 10 2 9 4 10 1 5 1 8 1 3 4 8 1 10 2 10 6 10 2 6 2 6	16 14 12 11 17 17 19 14 19 14 17 17 12 16 16 6 6

## Задача R. Широкая перестановка

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Дана перестановка  $p$  чисел от 1 до  $n$  и число  $w$ . Можно сколько угодно раз выполнить следующую операцию:

- Выбрать два индекса  $i$  и  $j$ , такие что  $i + w \leq j$  и  $|p_i - p_j| = 1$ .
- Поменять местами  $p_i$  и  $p_j$ .

Найдите лексикографически минимальную перестановку  $p$ , которую можно получить.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $w$  ( $1 \leq n \leq 500\,000$ ,  $1 \leq w \leq n - 1$ ) — размер перестановки и заданный параметр.

Вторая строка содержит  $n$  целых чисел  $p_1, \dots, p_n$  ( $1 \leq p_i \leq n$ ) — элементы перестановки. Гарантируется, что все  $p_i$  различны.

### Формат выходных данных

Выведите лексикографически минимальную перестановку, которую можно получить из заданной.

### Примеры

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