

## Задача А. Весело смотреть на фейерверки

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

На главной улице города устраивают фестиваль. Улица поделена на  $n$  участков, пронумерованных от 1 до  $n$  слева направо. Расстояние от любого участка до соседнего равно 1.

На фестивале запустят  $m$  фейерверков. При этом,  $i$ -ый ( $1 \leq i \leq m$ ) запуск придется на время  $t_i$  в участке  $a_i$ . Если вы будете в участке  $x$  ( $1 \leq x \leq n$ ) во время  $i$ -ого запуска, вы получите счастья в размере  $b_i - |a_i - x|$  (обратите внимание, что значение счастья может быть отрицательным).

За единичный промежуток времени вы можете пройти до  $d$  единиц длины, но сходить с главной улицы запрещено. В начальный момент времени (момент времени 1) вы можете оказаться на произвольном участке, ваша цель — максимизировать сумму полученного от просмотра фейерверков счастья. Найдите наибольшее суммарное счастье.

Обратите внимание, что в одно время могут запускаться два или несколько фейерверков.

### Формат входных данных

В первой строке записано три целых числа  $n, m, d$  ( $1 \leq n \leq 150000$ ;  $1 \leq m \leq 300$ ;  $1 \leq d \leq n$ ).

В каждой из следующих  $m$  строк записаны целые числа  $a_i, b_i, t_i$  ( $1 \leq a_i \leq n$ ;  $1 \leq b_i \leq 10^9$ ;  $1 \leq t_i \leq 10^9$ ). В  $i$ -ой строке записано описание  $i$ -ого запуска.

Гарантируется, что условие  $t_i \leq t_{i+1}$  ( $1 \leq i < m$ ) будет удовлетворено.

### Формат выходных данных

Выведите единственное целое число — максимальную сумму счастья, которую можно получить от просмотра всех фейерверков.

Пожалуйста, не используйте спецификатор `%lld` для чтения или записи 64-битных чисел на C++. Рекомендуется использовать потоки `cin, cout` или спецификатор `%I64d`.

### Примеры

стандартный ввод	стандартный вывод
50 3 1 49 1 1 26 1 4 6 1 10	-31
10 2 1 1 1000 4 9 1000 4	1992

## Задача В. Столбы

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

Сурок нашел ряд из  $n$  столбов. Высота  $i$ -ого столба равняется  $h_i$  метрам. Начиная с некоторого столба  $i_1$ , Сурок хочет пропрыгать по столбам  $i_2, \dots, i_k$ . ( $1 \leq i_1 < i_2 < \dots < i_k \leq n$ ). Сурок может перепрыгнуть со столба  $i$  на столб  $j$ , только если  $i < j$  и  $|h_i - h_j| \geq d$ , где  $|x|$  — абсолютное значение числа  $x$ .

Сурок просит вас найти последовательность прыжков максимальной длины и вывести её.

### Формат входных данных

В первой строке записано два целых числа  $n$  и  $d$  ( $1 \leq n \leq 10^5$ ,  $0 \leq d \leq 10^9$ ).

Во второй строке записано  $n$  чисел  $h_1, h_2, \dots, h_n$  ( $1 \leq h_i \leq 10^{15}$ ).

### Формат выходных данных

В первой строке должно быть записано одно целое число  $k$  — максимальная длина последовательности прыжков.

Во второй строке должно быть записано  $k$  целых чисел  $i_1, i_2, \dots, i_k$  ( $1 \leq i_1 < i_2 < \dots < i_k \leq n$ ) — номера столбов из максимально длинной последовательности прыжков.

Если есть несколько последовательностей прыжков максимальной длины, выведите любую из них.

### Примеры

стандартный ввод	стандартный вывод
5 2 1 3 6 7 4	4 1 2 3 5
10 3 2 1 3 6 9 11 7 3 20 18	6 1 4 6 7 8 9

### Замечание

В первом примере Сурок выбирает столбы 1, 2, 3, 5 с высотами 1, 3, 6, 4. Ещё одна последовательность прыжков длины 4 такова: 1, 2, 4, 5.

## Задача С. Панорама города

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Алиса в Нью-Йорке! Она хочет сделать несколько снимков знаменитых небоскребов и отправить набор фотографий Бобу. Алиса хочет составить наиболее красивый набор фотографий, и в этом ей требуется ваша помощь.

В городе  $n$  зданий, расположенных в линию слева направо,  $i$ -е из них имеет положительную высоту  $h_i$ . Высоты всех  $n$  зданий в городе различны. Кроме того, у каждого здания есть параметр красоты  $b_i$ . Красота зданий может быть положительна или отрицательна (или равна нулю), потому что некоторые здания портят вид.

Набор фотографий состоит из одной или более фотографий зданий. На каждом фото должно быть изображено одно или несколько зданий, расположенных подряд. Каждое здание должно попасть на **ровно одно** фото. Другими словами, если есть здание, которое не попало ни на одно фото, или присутствует более чем на одном фото, то такой набор фотографий считается некорректным.

Красота одной фотографии равняется красоте  $b_i$  самого низкого здания на фотографии. Общая красота набора фотографий равна сумме красот всех фотографий в наборе. Помогите Алисе определить максимально возможную красоту корректного набора фотографий.

### Формат входных данных

Первая строка содержит одно целое число  $n$  ( $1 \leq n \leq 3 \cdot 10^5$ ) — количество зданий.

Вторая строка содержит  $n$  различных целых чисел  $h_1, h_2, \dots, h_n$  ( $1 \leq h_i \leq n$ ), где  $i$ -е число равно высоте  $i$ -го здания.

Третья строка содержит  $n$  целых чисел  $b_1, b_2, \dots, b_n$  ( $-10^9 \leq b_i \leq 10^9$ ), где  $i$ -е число равно красоте  $i$ -го здания.

### Формат выходных данных

Выведите одно число — максимальную красоту, которую может иметь корректный набор фотографий.

### Примеры

стандартный ввод	стандартный вывод
5 1 2 3 5 4 1 5 3 2 4	15
5 1 4 3 2 5 -3 4 -10 2 7	10

### Замечание

В первом примере максимальная красота достигается, если Алиса сделает пять фотографий, каждая из которых содержит одно здание.

Во втором примере Алиса может достичь максимальной красоты набора, равной 10, если сделает четыре фотографии: три содержат одно здание каждая (здания 1, 2 и 5), их красоты равны  $-3$ , 4 и 7 соответственно, и еще одна фотография содержит здания 3 и 4, ее красота равна 2.

В третьем примере Алиса может просто сделать одно фото всего города.

В четвертом примере Алиса может сделать следующие фотографии для достижения максимальной красоты набора: отдельно фотографии зданий 1, 2, 8, 9 и 10, а также одно фото зданий 3, 4, 5, 6 и 7 вместе.

## Задача D. Гонки в Линейном Королевстве

Имя входного файла:	<code>stdin</code>
Имя выходного файла:	<code>stdout</code>
Ограничение по времени:	1 second
Ограничение по памяти:	256 megabytes

Вы — организатор гонок, и вы хотите провести серию заездов в Линейном Королевстве.

В Линейном Королевстве  $n$  дорог, пронумерованных от 1 до  $n$ . Дороги идут одна за другой слева направо, в порядке возрастания номера. Вам поступило несколько предложений проведения заездов. Для каждого заезда известно, на каких дорогах предлагается его проводить — некоторый набор **последовательных** дорог Королевства. Также известно ваше вознаграждение, в случае если вы решите провести данный заезд. Все заезды предлагается проводить в разное время, то есть одна и та же дорога может использоваться в нескольких заездах.

К несчастью, некоторые дороги в плохом состоянии, и их нужно чинить. Для каждой дороги известна стоимость ее ремонта. Заезд можно проводить только если отремонтированы все дороги, на которых проходит гонка. Ваша задача — отремонтировать такой набор дорог (возможно, все, или ни одной), чтобы ваша прибыль была как можно больше. Прибыль считается как сумма вознаграждений, полученных за проведение заездов, минус сумма денег, потраченных на ремонт дорог. Заметьте, что вы можете не чинить дороги вообще и получить нулевую прибыль.

Найдите максимальную возможную прибыль.

### Формат входных данных

В первой строке через пробел записано два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 2 \cdot 10^5$ ) — количество дорог и предложенных заездов соответственно.

Далее следует  $n$  строк. Каждая строка содержит одно целое неотрицательное число, не превосходящее  $10^9$  — стоимость починки очередной дороги. Стоимости починки даны в порядке начиная с дороги номер 1 заканчивая дорогой номер  $n$ .

Далее следует  $m$  строк по три числа в каждой:  $lb$ ,  $ub$ , и  $p$  ( $1 \leq lb \leq ub \leq n, 1 \leq p \leq 10^9$ ). Каждая строка описывает очередной заезд: предлагается использовать дороги с  $lb$  по  $ub$  включительно, и вознаграждение будет составлять  $p$ .

### Формат выходных данных

Выведите одно целое число — максимальную возможную прибыль, которую вы можете получить.

Пожалуйста, не используйте спецификатор `%lld` для чтения и записи 64-битных целых чисел в C++. Рекомендуется использовать потоки `cin`, `cout` (также вы можете использовать спецификатор `%I64d`).

## Примеры

stdin	stdout
7 4 3 2 3 2 1 2 3 1 2 5 2 3 5 3 5 3 7 7 5	4
2 1 0 3 1 2 5	2
3 1 10 10 10 1 3 10	0

## Замечание

В первом примере выгодно чинить дороги 1, 2, 3 и 7. Три заезда приносят доход 15. Ремонт стоит 11, значит прибыль равна 4.

## Задача E. Ezzat и таблица

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Моамен нарисовал таблицу из  $n$  строк и  $10^9$  столбцов, состоящую из цифр 0 и 1. Ezzat заметил, что Моамен рисует и заинтересовался, какое минимальное число строк нужно удалить, чтобы таблица стала прекрасной.

Таблица прекрасна, если и только если для каждых двух последовательных строк найдется хотя бы один столбец, содержащий 1 в этих двух строках.

Ezzat опишет таблицу с помощью числа строк  $n$ , и  $m$  отрезков таблицы, содержащих 1. Каждый отрезок задается тремя числами  $i$ ,  $l$ , и  $r$ , где  $i$  задает номер строки, а  $l$  и  $r$  задают первый и последний столбцы отрезка в этой строке.

Например, если  $n = 3$ ,  $m = 6$ , и заданы отрезки  $(1, 1, 1)$ ,  $(1, 7, 8)$ ,  $(2, 7, 7)$ ,  $(2, 15, 15)$ ,  $(3, 1, 1)$ ,  $(3, 15, 15)$ , то таблица выглядит так:

```
10000011000000000.....
000000100000000100.....
100000000000000100.....
```

Ваша задача — сказать Ezzat, сколько минимум строк нужно удалить, чтобы таблица стала прекрасной.

### Формат входных данных

В первой строке содержатся два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 3 \cdot 10^5$ ).

В каждой из следующих  $m$  строк содержатся три целых числа  $i$ ,  $l$  и  $r$  ( $1 \leq i \leq n$ ,  $1 \leq l \leq r \leq 10^9$ ). Каждая из этих  $m$  строк означает, что строка  $i$  содержит 1 в столбцах с  $l$  по  $r$ .

Обратите внимание, что отрезки могут накладываться.

### Формат выходных данных

В первой строке выведите одно целое число  $k$  — минимальное число строк, которые необходимо удалить.

Во второй строке выведите  $k$  различных чисел  $r_1, r_2, \dots, r_k$ , задающих строки, которые нужно удалить ( $1 \leq r_i \leq n$ ), в любом порядке.

Если существует несколько решений, выведите любое из них.

## Примеры

стандартный ввод	стандартный вывод
3 6 1 1 1 1 7 8 2 7 7 2 15 15 3 1 1 3 15 15	0
5 4 1 2 3 2 4 6 3 3 5 5 1 1	3 1 4 5
3 4 1 2 3 2 3 6 3 4 8 1 7 9	0

## Замечание

Таблица из первого примера приведена в условии задачи. Для нее уже верно, что:

- 1-я и 2-я строки имеют 1 в столбце 7.
- 2-я и 3-я строки имеют 1 в столбце 15.

Значит, исходная таблица уже прекрасная и мы не должны удалять ничего.

Во втором примере таблица следующая:

01100000.....

00011100.....

00111000.....

00000000.....

10000000.....

## Задача F. Выставка монет

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.2 секунд
Ограничение по памяти:	256 мегабайт

Аркадий и Кирилл пришли на выставку редких монет. Все монеты расположены в один ряд и пронумерованы слева направо от 1 до  $k$ , причем некоторые монеты лежат вверх аверсом (главной стороной), а некоторые — вверх реверсом (другой стороной).

Аркадий и Кирилл сделали несколько фотографий монет каждый, причем на каждой фотографии изображены верхние стороны нескольких подряд идущих монет. Аркадий интересуется аверсами, поэтому на каждом снимке, который сделал он, есть хотя бы одна монета, лежащая аверсом вверх. Кирилл, наоборот, интересуется реверсами, поэтому на каждом его снимке есть хотя бы одна монета, лежащая вверх реверсом.

Теперь снимки утеряны, и ребята лишь помнят границы отрезков монет, попавших на каждый из снимков. По данной информации о снимках вычислите остаток от деления на  $10^9 + 7$  количества способов выбрать для каждой монеты сторону, которой она будет лежать вверх, так, чтобы на каждом снимке Аркадия была бы хотя бы одна монета вверх аверсом, а на каждом снимке Кирилла была хотя бы одна монета вверх реверсом.

### Формат входных данных

Первая строка содержит три целых числа  $k$ ,  $n$  и  $m$  ( $1 \leq k \leq 10^9$ ,  $0 \leq n, m \leq 10^5$ ) — общее число монет, количество снимков Аркадия и Кирилла, соответственно.

Следующие  $n$  строк содержат описания снимков Аркадия, по одному в строке. Каждая из этих строк содержит два целых числа  $l$  и  $r$  ( $1 \leq l \leq r \leq k$ ), означающие, что среди монет с  $l$ -й по  $r$ -ю должна быть хотя бы одна вверх аверсом.

Следующие  $m$  строк содержат описания снимков Кирилла, по одному в строке. Каждая из этих строк содержит два целых числа  $l$  и  $r$  ( $1 \leq l \leq r \leq k$ ), означающие, что среди монет с  $l$ -й по  $r$ -ю должна быть хотя бы одна вверх реверсом.

### Формат выходных данных

Выведите одно целое число — количество способов выбрать для каждой монеты, какой стороной она будет лежать вверх, по модулю  $10^9 + 7 = 1000000007$ .

## Примеры

стандартный ввод	стандартный вывод
5 2 2 1 3 3 5 2 2 4 5	8
5 3 2 1 3 2 2 3 5 2 2 4 5	0
60 5 7 1 3 50 60 1 60 30 45 20 40 4 5 6 37 5 18 50 55 22 27 25 31 44 45	732658600

## Замечание

В первом примере возможны следующие расположения монет («А» — аверс, «Р» — реверс):

- АРААР,
- АРАРА,
- АРААР,
- РРААР,
- РРАРА,
- РРААР,
- АРРАР,
- АРРРА.

Во втором примере данная информация противоречива: согласно снимкам, вторая монета должна лежать одновременно аверсом и реверсом вверх, что невозможно. Значит, ответ равен 0.