

Задача А. 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 В. 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 каждого 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.....
00000010000000100.....
10000000000000100.....
```

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. 1-я и 2-я строки имеют 1 в столбце 7.
2. 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 С. 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,
- АРАРР,
- РРААР,
- РРАРА,
- РРАРР,
- 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.

Задача D. Уничтожить жуков

Имя входного файла:	stdin
Имя выходного файла:	stdout
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 megabytes

У преуспевающего фермера Анатолия есть несколько картофельных полей, приносящих ему стабильный и весьма неплохой доход. Однако ничего в этом мире не дается даром, в том числе и хороший урожай картофеля. Особенно тяжело защищать поля от колорадского жука. Раньше Анатолий привлекал для уничтожения жуков гастарбайтеров из соседней страны, но теперь их депортировали на родину. Поэтому Анатолий поехал в город и приобрел там новейшее чудо техники — робота, уничтожающего колорадского жука. Теперь все поля должны быть обработаны этим роботом.

Всего у Анатолия n картофельных полей, некоторые из которых соединены дорогами. По каждой дороге робот может передвигаться в обе стороны, причем на проезд по i -й из них он тратит d_i часов. Сеть дорог устроена таким образом, что существует ровно один способ добраться от любого поля до любого другого.

Робот должен обойти все поля, побывав на каждом хотя бы один раз. Каждое поле должно быть обработано роботом в одно из посещений этого поля. Обработка i -го поля занимает непрерывный отрезок времени длиной t_i часов. Робот начинает свой обход в момент времени 0 с поля номер 1 и должен вернуться на него после обработки всех полей. Кроме того, есть дополнительное ограничение: по каждой дороге робот может пройти только один раз в каждом из направлений.

Анатолию осталось только записать в память робота информацию о том, в каком порядке обходить и обрабатывать поля. Положение осложняется тем, что все поля неравнозначны: какие-то больше по размеру, какие-то больше поражены жуком и т.п. Поэтому разные программы действий для робота будут иметь разную эффективность. Анатолий оценил, что если обработка поля с номером i будет закончена в момент времени t , то его потери на этом поле составят $k_i \cdot t$ рублей. Помогите Анатолию задать роботу такую программу, чтобы минимизировать суммарные потери.

Формат входных данных

В первой строке записано целое число n ($2 \leq n \leq 10^5$). Во второй строке записано n целых чисел t_i ($1 \leq t_i \leq 10^4$). В третьей строке записано n целых чисел k_i ($1 \leq k_i \leq 10^4$). Следующие $n - 1$ строка описывают дороги: в i -й из этих строк содержится три целых числа: номера полей u_i и v_i , соединенных дорогой, и время проезда робота по дороге d_i ($1 \leq u_i, v_i \leq n, 1 \leq d_i \leq 10^4$).

Формат выходных данных

В первой строке выведите минимальную возможную сумму потерь. Далее выведите $3n - 2$ строки, описывающие действия робота. Для команды перемещения в i -е поле « $M\ i$ », для команды обработки i -го поля выводите « $P\ i$ ». Если оптимальных решений несколько, выведите любое из них.

Примеры

stdin	stdout
3 2 3 4 1 1 1 1 2 1 1 3 1	20 P 1 M 2 P 2 M 1 M 3 P 3 M 1
3 2 3 4 1 10 1 1 2 1 1 3 1	59 M 2 P 2 M 1 P 1 M 3 P 3 M 1

Задача Е. Дерево на Манхеттене

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

Участникам, использующим язык Python3, рекомендуется отправлять решения на проверку с использованием интерпретатора PyPy3.

Однажды мальчик Вова нашел корневое дерево[†] из n вершин, i -я вершина которого была соединена со своим предком p_i ($p_i < i$) ребром, с записанным на нём числом c_i . Корнем является вершина 1. Увидев его, Вова сразу же придумал интереснейшую задачу, которую вам предстоит решить.

Выкладыванием дерева на прямую назовём сопоставление каждой вершине v целого числа x_v ($1 \leq x_v \leq n$), при котором разным вершинам соответствуют разные числа, а каждое поддерево является последовательным подотрезком.

Проще говоря, выкладывание — это такая перестановка[‡] на вершинах, что для любой вершины v , если отсортировать значения x_u всех вершин u в поддереве вершины v , то образуется отрезок целых чисел $[l, r]$ для каких-то $1 \leq l \leq r \leq n$.

Стоимостью выкладывания назовём величину $\sum_{v=2}^n |x_v - x_{p_v}| \cdot c_v$.

Посчитайте минимальную стоимость выкладывания для данного вам дерева.

[†] *Дерево* — это связный неориентированный граф без циклов. *Корневое дерево* — дерево с выделенной вершиной, которую называют корнем.

[‡] *Перестановкой* длины n является массив, состоящий из n различных целых чисел от 1 до n в произвольном порядке. Например, $[2, 3, 1, 5, 4]$ — перестановка, но $[1, 2, 2]$ не перестановка (2 встречается в массиве дважды) и $[1, 3, 4]$ тоже не перестановка ($n = 3$, но в массиве встречается 4).

Формат входных данных

Первая строка содержит одно целое число n ($2 \leq n \leq 5000$) — количество вершин в дереве.

Затем следуют $n - 1$ строк, i -я из которых содержит два целых числа p_{i+1} и c_{i+1} ($1 \leq p_{i+1} \leq i, 0 \leq c_{i+1} \leq 10^{11}$) — предок $(i + 1)$ -й вершины и число, записанное на ребре из $i + 1$ в p_{i+1} .

Формат выходных данных

В единственной строке выведите одно число — минимальную стоимость выкладывания данного вам дерева.

Примеры

стандартный ввод	стандартный вывод
5 1 6 1 5 2 4 2 3	21
8 1 6 1 9 2 2 2 9 5 4 3 9 6 11	56

Замечание

Можно показать, что добиться на первом тесте из условия стоимости выкладывания меньше, чем 21 нельзя. Данная стоимость получается с помощью выкладывания $x = \{4, 3, 5, 2, 1\}$.

Аналогично, для второго теста стоимости выкладывания меньше, чем 56 получить нельзя. Пример выкладывания с данной стоимостью $x = \{6, 5, 7, 1, 4, 3, 8, 2\}$.

Задача F. Карен и супермаркет

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

На пути домой Карен решила зайти в супермаркет за продуктами.



Ей нужно купить много продуктов, но, так как она еще учится, ее бюджет ограничен. Она может потратить не более b долларов.

В супермаркете есть n товаров. Товар номер i может быть куплен за c_i долларов. Конечно, каждый товар может быть куплен только один раз.

Супермаркет хочет расширяться, поэтому Карен получила n купонов. Если Карен купит i -й товар, то она может использовать i -й купон для того, чтобы уменьшить цену этого товара на d_i . Конечно, купон не может быть использован без покупки соответствующего товара.

Однако, на использование купонов накладывается ограничение. Для всех $i \geq 2$, чтобы использовать купон номер i , Карен должна использовать также купон номер x_i (для использования которого может понадобиться использовать другие купоны, чтобы выполнить условие).

Карен хочет узнать, сколько максимум товаров она может купить, не превысив своего бюджета b ?

Формат входных данных

Первая строка содержит два целых числа n и b ($1 \leq n \leq 5000$, $1 \leq b \leq 10^9$) — число товаров в магазине и количество денег у Карен, соответственно.

Следующие n строк описывают товары. А именно:

- Строка номер i начинается с двух целых чисел c_i и d_i ($1 \leq d_i < c_i \leq 10^9$) — цены i -го товара и скидка при использовании купона для i -го товара, соответственно.
- Если $i \geq 2$, то далее следует еще одно целое число x_i ($1 \leq x_i < i$), означающее, что перед тем как будет использован этот купон, необходимо использовать купон номер x_i .

Формат выходных данных

Выведите одно целое число: количество товаров, которые Карен может купить, не превысив бюджет.

Примеры

стандартный ввод	стандартный вывод
6 16 10 9 10 5 1 12 2 1 20 18 3 10 2 3 2 1 5	4
5 10 3 1 3 1 1 3 1 2 3 1 3 3 1 4	5

Задача G. Мисс Punyverse

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

Дуб имеет n гнезд, пронумерованных целыми числами от 1 до n . Гнездо i это дом для b_i пчел и w_i ос.

Некоторые гнезда соединены ветками. Мы называем два гнезда соседними если существует ветка, соединяющая их. Простой путь из гнезда x в y задается последовательностью $s_0, \dots, s_p$ различных гнезд, где p это неотрицательное целое число и $s_0 = x$, $s_p = y$ и s_{i-1} и s_i соседние для всех $i = 1, \dots, p$. Ветки дуба расположены так, что для любой пары гнезд x и y существует единственный простой путь из x в y . По причине этого и биологи и программисты согласны, что дуб, по факту, является деревом.

Деревня это непустое множество V гнезд, такое что для всех x и y из V , существует простой путь из x в y , все гнезда вдоль которого также лежат в V .

Множество деревень $\mathcal{P}$ называется разбиением если каждое из n гнезд содержится в ровно одной деревне из $\mathcal{P}$. Другими словами, никакие две деревни из $\mathcal{P}$ не содержат одно и то же гнездо и в объединении дают все n гнезд.

На дубе проводится ежегодный конкурс красоты мисс Punyverse. Всего два участника участвуют в этом году, это некрасивая оса и красивая пчела. Победитель конкурса красоты определяется голосованием, систему которого мы сейчас опишем. Пусть $\mathcal{P}$ это разбиение гнезд на m деревень $V_1, \dots, V_m$. Проводятся локальные выборы в каждой деревне. Каждое насекомое в деревне голосует за свое любимое насекомое. Если в деревне окажется, что строго больше голосов отдано за некрасивую осу чем за красивую пчелу, тогда некрасивая оса объявляется победителем в этой деревне. Иначе, красивая пчела побеждает.

Известно, что пчелы всегда голосуют за пчел, то есть проголосуют за красивую пчелу в этом году и осы всегда голосуют за ос, то есть проголосуют за некрасивую осу в этом году. Все насекомые в улье участвуют в голосовании.

Мэр Waspacito, а также его заместитель Alexwasp, хотят, чтобы некрасивая оса победила. Они имеют влияние и могут выбрать разбиение дуба на ровно m деревень. Если они выберут разбиение оптимально, определите максимальное количество деревень, в которых победит некрасивая оса.

Формат входных данных

В первой строке находится единственное целое число t ($1 \leq t \leq 100$), количество тестовых случаев. В следующих строках находится описание тестовых случаев.

В первой строке каждого тестового случая находится два целых числа n и m ($1 \leq m \leq n \leq 3000$). Во второй строке находится n целых чисел $b_1, b_2, \dots, b_n$ ($0 \leq b_i \leq 10^9$). В третьей строке находится n целых чисел $w_1, w_2, \dots, w_n$ ($0 \leq w_i \leq 10^9$). В следующих $n - 1$ находятся описания соседних пар гнезд, i -я из них содержит два целых числа x_i и y_i ($1 \leq x_i, y_i \leq n$, $x_i \neq y_i$), обозначающих номера гнезд, соединенных веткой. Гарантируется, что эти пары задают дерево.

Гарантируется, что сумма n по всем тестовым случаям не превосходит 10^5 .

Формат выходных данных

Для каждого тестового случая выведите целое число, равное максимальному количеству деревень, в которых победит некрасивая оса по всем разбиениям дуба на m деревень.

[illegible]

Замечание

В первом тестовом случае, нужно разделить $n = 4$ гнезд на $m = 3$ деревень. Мы можем так разделить, что некрасивая оса победит в 2 деревнях, например с помощью такого разбиения: $\{\{1, 2\}, \{3\}, \{4\}\}$. В этом разбиении:

- Некрасивая оса побеждает в деревне $\{1, 2\}$, получив 181 голосов тогда как красивая пчела получила 170 голосов;
- Некрасивая оса также побеждает в деревне $\{3\}$, получив 111 голосов тогда как красивая пчела получила 70 голосов;
- Некрасивая оса проигрывает в деревне $\{4\}$, получив 0 голосов тогда как красивая пчела получила 50.

Таким образом некрасивая оса победила в 2 деревнях и можно показать, что это максимально возможное число.

Во втором тесте мы должны разделить $n = 2$ гнезд на $m = 1$ деревень. Есть только один способ это сделать: $\{\{1, 2\}\}$. В этом разбиении некрасивая оса получит 563 голосов и красивая пчела также получит 563 голосов. Некрасивая оса побеждает, если получает строго больше голосов. Поэтому некрасивая оса не выиграет ни в какой деревне.

Задача Н. Покраска дерева

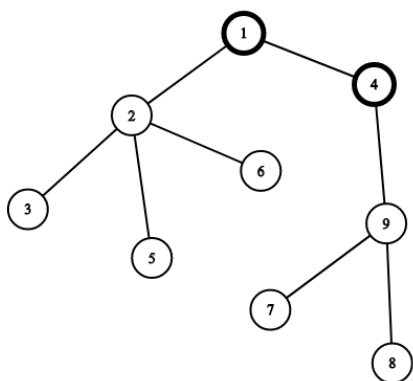
Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вам задано дерево (неориентированный связный ациклический граф), состоящее из n вершин. Вы играете в игру на этом дереве.

Изначально все вершины белые. На первом ходу игры Вы выбираете одну вершину и красите ее в черный. Затем на каждом ходу Вы выбираете белую вершину, смежную (соединенную ребром) с **любой** черной вершиной и красите ее в черный.

Каждый раз, когда вы выбираете вершину (даже во время первого хода), Вы получаете количество очков, равное размеру компоненты связности, состоящей только из белых вершин, содержащей выбранную вершину. Игра заканчивается, когда все вершины покрашены в черный цвет.

Рассмотрим следующий пример:



Вершины 1 и 4 уже покрашены в черный цвет. Если Вы выберете вершину 2, Вы получите 4 очка за компоненту связности, состоящую из вершин 2, 3, 5 и 6. Если Вы выберете вершину 9, Вы получите 3 очка за компоненту связности, состоящую из вершин 7, 8 и 9.

Ваша задача — максимизировать количество очков, которое Вы получите.

Формат входных данных

Первая строка содержит одно целое число n — количество вершин в дереве ($2 \leq n \leq 2 \cdot 10^5$).

Каждая из следующих $n - 1$ строк описывает ребро дерева. Ребро i описывается двумя целыми числами u_i и v_i , номерами вершин, которые оно соединяет ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$).

Гарантируется, что заданные ребра образуют дерево.

Формат выходных данных

Выведите одно целое число — максимальное количество очков, которое вы получите, если будете играть оптимально.

Примеры

стандартный ввод	стандартный вывод
9 1 2 2 3 2 5 2 6 1 4 4 9 9 7 9 8	36
5 1 2 1 3 2 4 2 5	14

Замечание

Первый тестовый пример показан в условии задачи.

Задача I. ДП на дереве. Или нет?

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 1024 мегабайта

Дано неориентированное дерево с n вершинами. Маленький Q хочет удалить все ребра дерева. Сначала он выбирает начальную вершину. В каждой операции он может выполнить одно из следующих трех действий:

1. Выбрать смежное не удаленное ребро, пройти по нему и удалить это ребро.
2. Заплатить стоимость A , чтобы восстановить удаленное ребро.
3. Заплатить стоимость B , чтобы телепортироваться к любой вершине.

Найдите минимальную общую стоимость, необходимую для удаления всех ребер дерева.

Формат входных данных

Первая строка содержит три положительных целых числа: n , A и B ($1 \leq n \leq 5 \cdot 10^5$; $1 \leq A, B \leq 10^9$).

Каждая из следующих $n - 1$ строк содержит два целых числа u и v : две вершины, соединенные ребром ($1 \leq u, v \leq n$).

Формат выходных данных

Выведите одно целое число: минимальная общая стоимость, необходимая для удаления всех ребер дерева.

Примеры

стандартный ввод	стандартный вывод
5 100 1000 1 2 2 3 3 4 4 5	0
5 100 200 1 2 1 3 2 4 2 5	100

Задача J. Гроб или халява?

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	1024 мегабайта

В прекрасном городе УСРС всего n домов, расположенных на равных интервалах вдоль прямой линии. Жители дома i требуют, чтобы скорость сетевого соединения в их доме была не менее a_i . Новый мэр, Дохун, намерен установить новые антенны, чтобы обеспечить стабильное сетевое соединение для граждан.

Когда устанавливается антенна с мощностью x , она обеспечивает скорость сетевого соединения $x - \ell + 1$ для каждого из ℓ последовательных домов. Здесь количество домов ℓ , которые получают сетевое соединение, можно установить непосредственно как целое число, не превышающее x , при установке антенны, и также можно установить разные значения для каждой антенны. Однако из-за технических ограничений максимальная мощность одной антенны составляет P , и чтобы предотвратить радиопомехи, антенны должны быть установлены таким образом, чтобы ни две антенны не обеспечивали сеть одному и тому же дому.

Чтобы удовлетворить всех граждан, Дохун хочет обеспечить необходимую скорость сетевого соединения в каждом доме. Кроме того, чтобы сэкономить бюджет города, он стремится минимизировать сумму мощностей установленных антенн. Помогите Дохуну найти минимальную сумму мощностей антенн, необходимую для обеспечения необходимых скоростей сетевого соединения в каждом доме.

Формат входных данных

Первая строка содержит два целых числа, n и p : количество домов в городе и максимальная мощность одной антенны ($1 \leq n \leq 5 \cdot 10^5$; $1 \leq P \leq 10^9$).

Вторая строка содержит n целых чисел $a_1, a_2, \dots, a_n$: требуемая скорость сетевого соединения в каждом доме ($1 \leq a_i \leq p$).

Формат выходных данных

Выведите минимальную сумму мощностей антенн, необходимую для обеспечения необходимых скоростей сетевого соединения во всех домах.

Пример

стандартный ввод	стандартный вывод
7 5 2 4 3 3 1 4 5	19

Замечание

Установив:

- антенну с мощностью 5, которая обеспечивает скорость соединения 4 от дома 1 до дома 2,
- антенну с мощностью 5, которая обеспечивает скорость соединения 3 от дома 3 до дома 5,
- антенну с мощностью 4, которая обеспечивает скорость соединения 4 в доме 6, и
- антенну с мощностью 5, которая обеспечивает скорость соединения 5 в доме 7,

мы можем сделать общую сумму мощностей антенн равной 19, удовлетворяя требованиям всех домов.