

Задача А. Keep Talking and Nobody Explodes

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

Название задачи является ссылкой на изветную игру (мало ли кто не понял).

Это интерактивная задача.

Роберт Оппенгеймер разрабатывает готовится к испытанию «Тринити». Для этого ему необходимо очень точно настроить мощность ядерной бомбы, чтобы не произошли непредвиденные реакции. Максимальная мощность, при которой испытание будет успешным, равна p — при любом большем значении запуск бомбы, скорее всего, приведет к катастрофическим последствиям.

Однако значение p Роберту неизвестно. Известно только, что $2 \leq p < 10^{12}$. Для того, чтобы найти значение p , Роберт может провести ряд тестов, а также попросить совет у известных ученых, например, у Альберта Эйнштейна.

1. Чтобы попросить совет, Роберт может назвать любое целое число t до 10^6 , и получить в ответ информацию о том, лежит ли t в интервале от \sqrt{p} не включительно до p включительно.
2. Если Роберт проводит эксперимент, он также выбирает целое число t до 10^{12} , и
 - если $t > p$, эксперимент приводит к случайному срабатыванию тестового образца, в результате чего, скорее всего, много людей пострадают, а лаборатория будет разрушена;
 - если $t \leq p$, то эксперимент проводится успешно, и в результате эксперимента Роберт получает равномерно распределенное случайное число из отрезка $\left[\frac{p}{t}, \frac{p^2}{t^2}\right]$.

Сможете ли вы решить эту задачу не медленнее Оппенгеймера? В зависимости от p он умеет решать ее не более, чем за 20 запросов. Но поскольку Оппенгеймер все-таки был гением, вам разрешается сделать на два запроса больше.

Формат входных данных

В этой задаче нет входных данных.

Протокол взаимодействия

Ваш протокол взаимодействия с интерактором заключается в обмене запросами и ответами на них.

1. Для того, чтобы совершить запрос первого типа («попросить совет»), выведите «ADVICE t », где t — выбранное для этого совета число от 1 до 10^6 . В ответ на этот запрос интерактор вернет вам ответ «YES», если $\sqrt{p} < t \leq p$, и «NO» иначе.
2. Для того, чтобы провести эксперимент с мощностью t , выведите «EXPERIMENT t », где t должно лежать от 1 до 10^{12} включительно, и
 - если $t \leq p$, интерактор вернет вам ответ в формате «OK x », где x — равновероятно выбранное из отрезка $\left[\frac{p}{t}, \frac{p^2}{t^2}\right]$ число с плавающей точкой с хотя бы 18 знаками после точки;
 - иначе, интерактор напечатает «BOOM» и завершится с вердиктом WA (Wrong Answer).
3. Для того, чтобы сообщить, что вы выяснили значение числа p , выведите «SUCCESS p », указав определенное вами значение p . Тогда
 - если указанное значение верное, интерактор вернет вам ответ строку «OK» и завершится;

- иначе, интерактор напечатает «BOOM» и завершится с вердиктом WA (Wrong Answer).

После каждой выведенной строки не забывайте сбросить буфер вывода (`cout.flush()` в C++, `System.out.flush()` в Java и `sys.stdout.flush()` в Python). Если ваша программа не сбрасывает буфер вывода, либо не завершается после завершения интерактора (с любым вердиктом), она может получить вердикты TL или IL.

Если ваша программа совершает больше 22 запросов первого и второго типа, интерактор, как и в остальных случаях ошибки, выведет «BOOM» и завершится с вердиктом WA. Если же вы верно узнаете p за $a_p \leq 22$ запросов, но авторское решение делает $a_j < a_p - 2$ запросов, вы все равно получаете вердикт WA.

Примеры

стандартный ввод	стандартный вывод
NO OK 2.612136477990739891 OK 1.000000000000000000 OK	ADVICE 3 EXPERIMENT 1 EXPERIMENT 2 SUCCESS 2
OK 7390019438.576070785522460938 YES OK 1.836419809778194523 NO YES OK 1.001041327128544323 OK 1.000000000000000000 OK	EXPERIMENT 2 ADVICE 200000 EXPERIMENT 200000 ADVICE 31000 ADVICE 31100 EXPERIMENT 310500 EXPERIMENT 310771 SUCCESS 310771

Задача В. Кадровые перестановки

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

Наступает конечный этап проекта Манхеттен, поэтому Роберт Оппенгеймер планирует перевести всех сотрудников в одну команду для проверки расчетов и финальных приготовлений. Сейчас сотрудники работают в здании, в котором есть n больших залов с аппаратурой для исследований. В зале i находится ровно a_i сотрудников. При этом в здании залы стоят в линию, поэтому за раз сотрудник может перейти лишь в соседний по номеру зал.

Оппенгеймер понимает, что каждому работнику для работы требуется его оборудование, поэтому нельзя просто взять и привести всех в один зал: перевод сотрудника сопровождается переносом всего его оборудования, что довольно накладно и требует времени, которое можно было тратить на исследования. Поэтому начальство хочет сделать объединение команд наименее накладным. *Сложность* объединения считается как количество переходов сотрудников из своих залов в соседние по пути в зал, где они все по итогу будут работать.

Помогите найти минимальную возможную сложность процесса объединения. Высшее командование заранее выражает вам благодарность за сотрудничество.

Формат входных данных

В первой строке ввода дано целое число n — количество залов в здании проекта Манхэттен ($1 \leq n \leq 2 \cdot 10^5$).

В следующей строке через пробел перечислены n целых чисел a_i — количество сотрудников в зале i ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите одно число — минимальную сложность перевода всех сотрудников в один и тот же зал.

Примеры

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

Задача С. Барби в реальном мире

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

Барби смогла попасть из Барбилэнда в реальный мир!

Тут все устроено совсем по-другому: и социальные нормы, и вообще все! Особенно ей интересно наблюдать за тем, как быстро кукол Барби разбирают в магазинах с полок. Несмотря на то, что девочка Саша, которая играла конкретно с нашей Барби, считает, что такие куклы навязывают нереалистичные стандарты, в принципе куклы Барби все еще очень популярны, поэтому любая новая модель идет нарасхват.

Всего в магазине, в котором Барби сейчас находится, n расположенных подряд на одной полке рядов с куклами. В i -м ряду стоит ровно a_i кукол. За новой моделью, «IT-Барби», пришли m детей, и каждый хочет успеть взять как можно больше. Каждый ребенок забирает ровно по одной кукле в секунду из ряда, рядом с которым он стоит.

В какой-то момент может случиться так, что оставшееся количество кукол a в каком-то ряду меньше количества стоящих у этого ряда детей b . Тогда:

- какие-то b детей успевают в очередную секунду взять по одной кукле;
- оставшиеся $a - b$ детей **по очереди** и **мгновенно** перед взятием куклы перемещаются в ближайший ряд, в котором количество кукол больше, чем количество стоящих рядом детей;
- если какой-то ребенок не может найти ряд, в котором на него бы тоже хватило еще одной куклы Барби в следующую секунду, он расстраивается и уходит на кассу с тем количеством кукол, которое уже собрал.

Поскольку наша Барби из Барбилэнда хочет, чтобы дети были одинаково довольны приобретениями, она хочет перед началом всего этого процесса расставить детей около рядов кукол так, чтобы в конце у каждого ребенка было одинаковое количество кукол. Получится ли у нее?

Формат входных данных

В первой строке ввода даны два целых числа n и m — количество рядов и количество детей ($1 \leq n \leq 10^5$; $1 \leq m \leq 10^9$).

Во второй строке через пробел даны n целых чисел a_i — количество кукол в каждом ряду ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите «YES» (без кавычек), если можно расставить детей так, чтобы они получили одинаковое количество кукол, и «NO» иначе.

Примеры

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

Замечание

В первом примере можно поставить каждого ребенка к соответствующему ряду. После трех секунд первый ребенок переместится в третий ряд, спустя еще секунду куклы закончатся.

Задача D. Прелестная рассадка

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

Сегодня у Барби гости. Она уже подготовила чай, печенье, круглый стол и n стульев, ровно под число гостей.

Барби считает, что рассадка будет прелестной, если гость i будет сидеть на стульях с l_i места по r_i . Однако, прежде чем все распланировать, она забыла учесть, возможна ли вообще такая рассадка. Поэтому Барби теперь в срочном порядке, пока не прибыли гости, хочет узнать, сможет ли она рассадить всех гостей прелестным образом, или же ей придётся заново придумывать прелестную рассадку.

Времени у нее мало, так что ей понадобится ваша помощь! Определите, существует ли прелестная рассадка в текущих условиях, и, если да — найдите хотя бы одну такую.

Формат входных данных

В первой строке дано одно целое число n — количество гостей Барби ($1 \leq n \leq 10^5$).

В следующих n строках даны пары целых чисел l_i, r_i — номера мест для гостя i ($1 \leq l_i \leq r_i \leq n$).

Формат выходных данных

В первой строке выведите «YES», если Барби сможет рассадить всех гостей так, чтобы рассадка была прелестной, и «NO» иначе.

Если ответ на задачу «YES», во второй строке выведите n чисел a_i — описание рассадки гостей, где a_i — номер гостя ($1 \leq a_i \leq n$), которого нужно посадить на место i .

Примеры

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

Задача Е. Цепная реакция

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

Роберт Оппенгеймер любит представлять взаимодействия между ядрами атомов в ядерной реакции в виде деревьев.

Для каждого ядра, используемого Робертом, известно две характеристики: a_i и b_i — количество нейтронов и протонов. При взаимодействии ядра i с ядром j происходит переход заряда от ядра i к ядру j . При этом значение заряда изменяется одним из двух способов: увеличивается на $a_j - a_i$ или увеличивается на $b_j - b_i$.

Во время очередного эксперимента Оппенгеймер передал заряд, равный 1, ядру с номером s . Далее этот заряд распространился по дереву взаимодействий по всем остальным ядрам так, что при каждом взаимодействии двух ядер, заряд изменился одним из двух описанных выше способов.

Ученый хочет понять, какое максимальное значение заряда может быть среди ядер в дереве взаимодействий. Помогите ему ответить на этот вопрос.

Формат входных данных

В первой строке входных данных даны два числа n и s — количество ядер атомов в дереве взаимодействий и номер атома, которому Роберт передал заряд в начале эксперимента ($1 \leq n \leq 10^5$; $1 \leq s \leq n$).

Во второй строке через пробел дано n чисел a_i — количество нейтронов в ядре i ($1 \leq a_i \leq 10^9$).

В третьей строке через пробел дано n чисел b_i — количество протонов в ядре i ($1 \leq b_i \leq 10^9$).

В следующих $n - 1$ строках дано описание возможных взаимодействий между ядрами атомов. Взаимодействие задается двумя числами — номерами ядер (от 1 до n), между которыми оно происходит.

Гарантируется, что граф взаимодействий связан и образует дерево.

Формат выходных данных

Выведите одно число — максимальный заряд, который может получить атом в результате проведения эксперимента.

Примеры

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

Замечание

В первом примере из входных данных на ядро атома 1 подается заряд 1. Опишем, какие максимальные заряды можно получить на каждом из ядер:

- Ядро 1: 1

- Ядро 2: $\max(-1, -2) = -1$
- Ядро 3: $\max(-2, -3) = -2$
- Ядро 4: $\max(-2, 2) = 2$
- Ядро 5: $\max(-2, 1) = 1$

Максимальное возможное значение заряда достигается у ядра 4 и это 2.

Задача F. Самый милый дом

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

Барби решила построить самый милый дом в Барбилэнде (благо, пока Кены развлекаются на пляже, она имеет серьезную и уважаемую работу, так что средства позволяют).

Затяя у нее правда странная, потому что идею она услышала от Странной Барби. Она собирается построить дом из n комнат, расположенных вплотную в ряд друг за другом, ведь «модный дом — длинный дом». Каждая комната имеет высоту a и ширину b , если смотреть на них сбоку, чтобы было видно весь ряд. Таким образом, ее дом будет иметь ширину ровно $b \cdot n$.

При чем у каждой комнаты есть свое ограничение по тому, на какой высоте можно ее расположить. Будет странно, если чердак будет стоять на земле, а ванная комната — витать в воздухе. По мнению Странной Барби, чтобы дом получился милым, i -я комната не может заходить ниже l_i или выше h_i над уровнем земли (то есть потолок не может быть выше h_i , а пол — ниже l_i).

Иными словами, если завести координаты, i -я комната может располагаться только внутри прямоугольника, ограниченного точками $((i-1) \cdot a, l_i)$ слева-снизу и $(i \cdot a, h_i)$ справа-сверху.

Также Барби понимает, что из комнат желательно иметь возможность проходить в соседние, не выходя на улицу. Поэтому она хочет отрегулировать все свои комнаты по высоте так, чтобы можно было построить непрерывный коридор, проходящий на одной высоте внутри комнат через как можно большее количество комнат подряд. Более формально, надо расположить прямоугольники комнат так, чтобы на занимаемом им пространстве можно было выбрать полосу высоты 1 максимальной возможной длины.

Помогите Барби построить модный дом с самым длинным возможным коридором!

Формат входных данных

В первой строке даны три целых числа n , a и b — количество комнат, а также высота и ширина каждой комнаты ($1 \leq n \leq 10^6$; $1 \leq a, b \leq 10^9$).

В i -й из следующих n строк даны два числа l_i и h_i — ограничения снизу и сверху на расположение i -й комнаты ($0 \leq l_i, h_i \leq 10^9$; $l_i + a \leq h_i$).

Формат выходных данных

Выведите единственное целое число — максимальную длину коридора, который получится построить в милом доме.

Примеры

стандартный ввод	стандартный вывод
3 3 4 4 8 6 10 3 6	8
2 2 2 1 4 3 7	4

Задача G. Прогулка с Барби

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

Пляжный Кен счастлив, потому что Стереотипная Барби наконец ответила ему взаимностью и согласилась вместе прогуляться по пляжу!

Пляж представляет собой прямоугольник $h \times w$. n клеток пляжа представляют из себя особые места. В некоторых из них лежат валуны и заходить на них нельзя, а в остальных лежат интересные вещицы — ракушки, кораллы и различного рода приятности, увидев которые, Барби увеличит свою симпатию к Кену на какую-то величину. Для каждой клетки известно, пустая она, заполнена валуном, или там лежит что-то интересное — то есть, при проходе по этой клетке (i, j) , симпатия Барби к Кену увеличится на $c_{i,j}$. Помогите Кену составить оптимальный маршрут прогулки, чтобы в конце симпатия Барби была максимальна.

Прогулка начинается в точке $(1, 1)$ и заканчивается в точке (h, w) . Так как у Барби не очень много времени, они могут гулять только влево, вправо и вниз по пляжу.

Формат входных данных

В первой строке входных данных дано три целых числа — h , w и n — высота и ширина пляжа и количество особенных мест ($1 \leq h \leq 10^5$; $1 \leq w \leq 10^9$; $1 \leq n \leq 10^5$).

В следующих n строках дано описание особых мест. Описание особого места представляется или как $ij + d$, если на пляже в строке i и столбце j стоит лежит вещь, приносящая $+d$ к симпатии, или $ij\#$, если в клетке i, j располагается валун ($1 \leq d \leq 10^9$).

Формат выходных данных

В единственной строке выведите ответ на задачу — максимальная симпатия, которой Кен может добиться от Барби в конце прогулки (изначально симпатия 0).

Пример

стандартный ввод	стандартный вывод
4 5 11 1 3 + 2 1 5 # 2 2 + 4 2 3 # 3 1 + 1 3 2 + 1 3 4 # 3 5 + 5 4 1 + 10 4 2 # 4 4 + 2	10

Задача Н. Конференция

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

Для того, чтобы делиться своими научными достижениями, ученые собираются на международные конференции. В том числе такие конференции проходят и у физиков-ядерщиков (и особенно популярны они начали становиться как-раз во времена Оппенгеймера).

Известно, что в каждом из n городов проживает некоторое, возможно нулевое, количество ученых-ядерщиков. Между некоторыми городами мира можно добраться на некотором транспорте за определенную стоимость. В этот раз Роберту Оппенгеймеру, собирающему конференцию, придется подумать над менее научным вопросом: а как оптимальнее собрать всех ученых в одном городе?

Чтобы собрать всех ученых в одном городе, для тех из них, кто не проживает в городе проведения конференции, необходимо оплатить им билеты для перемещения из их родного города в город проведения. Разумеется, средства, выделенные на конференцию, хочется потратить на что-то более полезное, чем билеты, поэтому город для проведения конференции Роберт хочет выбрать так, чтобы минимизировать суммарные затраты на сборы ученых.

Помогите ему определить, какое минимальное количество денег придется потратить на покупку всех билетов для всех ученых, если пока еще не поздно выбрать любой город проведения.

Формат входных данных

В первой строке входных данных даны два целых числа n и m — количество городов и способов перемещений напрямую между парами городов ($1 \leq n \leq 250$; $n - 1 \leq m \leq 4 \cdot 10^4$).

Во второй строке даны n чисел c_i — количество ученых, проживающих в каждом из городов ($0 \leq c_i \leq 10^7$).

В следующих m строках дано описание прямых маршрутов перемещений между городами. Описание каждого из них состоит из трех целых чисел u_i , v_i и w_i — пары номеров соединяемых городов и стоимости перемещения ($1 \leq u_i, v_i \leq n$; $u_i \neq v_i$; $1 \leq w_i \leq 10^7$).

Каждый описанный способ перемещения позволяет попасть из одного города в другой и наоборот. Гарантируется, что для любых двух городов есть не более одного способа переместиться между ними напрямую, но для любой пары городов есть хотя бы один (не обязательно прямой) маршрут между ними.

Формат выходных данных

В первой и единственной строке выведите целое число — минимальное количество денег, которое потребуется придется потратить на покупку всех билетов для всех ученых.

Примеры

стандартный ввод	стандартный вывод
4 4 1 2 2 3 1 2 3 1 3 1 2 3 6 2 4 1	14
5 8 1 3 1 1 2 2 5 5 4 5 10 4 3 3 3 2 6 2 1 5 5 1 6 3 5 2 4 2 10	28

Задача I. Интерпретация

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

Для любого достаточно мощного оружия и лабораторий, его обслуживающих, требуется программное обеспечение. Поскольку в эру Оппенгеймера компьютеры только зарождались, это утверждение еще не было правдой. Но это не означает, что программирования тогда вообще не существовало — любой повседневный алгоритм можно представить в виде псевдокода.

В лаборатории Оппенгеймера на стене висел псевдокод, выводящий какие-то характеристики ядерной бомбы. Псевдокод устроен довольно просто: он состоит только из блоков, которые мы будем обозначать как `<block>`, и работает только с целыми числами, поэтому Оппенгеймер при необходимости запускает его у себя в уме.

В этом псевдокоде:

1. `<block>` — либо примитивное действие, либо цикл;
2. Цикл имеет вид

```
for <var> = <value1>...<value2> {  
    <block>  
    <block>  
    ...  
}
```

Такой цикл проходит переменной `<var>` по всем значениям от `<value1>` до `<value2>` *включительно*, и для каждой выполняет указанную в своем теле последовательность блоков. Переменная `<var>` при этом может использоваться только внутри этого цикла и только с правой части от знака `'='`.

3. `<var>` — имя переменной;
4. `<value>` — либо имя переменной, либо целочисленная константа;
5. Примитивная операция может быть
 - либо `«print(<var>)»` — напечатать значение переменной,
 - либо `«read(<var>)»` — записать в переменную значение со ввода,
 - либо присвоение значения в переменную в формате `«<var> = <expression>»`
6. `<expression>` — либо имя переменной, либо целочисленная константа, либо арифметическое выражение;
7. Арифметические выражения имеют вид `«<value1> [+ -] <value2>»`, то есть либо сумма, либо разность двух величин.

Для этого псевдокода верно следующее:

- Счетчики циклов уникальные и не используются вне соответствующих циклов.
- Значения счетчиков циклов никогда не изменяются внутри цикла (не стоят слева от знака `'='` и не считываются с помощью `read`).
- Область видимости переменных — глобальная: если какая-то переменная создается, она после этого доступна до конца работы программы.
- Если верхняя граница цикла меньше нижней, цикл не совершает итерации вообще.

Вы — молодой лаборант, не разбирающийся в программировании (действительно, его пока почти что нет). Основные проблемы в понимании у вас вызывают вложенные циклы. Вы не понимаете вложенность, поэтому хотите перевести этот псевдокод на другой язык, чтобы иметь возможность помогать Роберту Оппенгеймеру в работе.

Другой язык поддерживает все те же примитивные операции (ввод, вывод и присвоение значений в переменную), и также имеет глобальную область видимости переменных, но в нем абсолютно нет скобок и возможности делать вложенные циклы. Вместо этого:

- Для вызова `print` или `read` надо написать соответствующую команду *заглавными буквами* и ее аргумент через пробел.
- Для записи значения в переменную надо написать `<var> GETS <expression>`.

А также в языке есть две специальные команды:

- Команда `MACRO`. Если вы напишете `MACRO <macro_name>:`, то в следующих строках с отступом в 4 пробела по одной команде на строке можно перечислить последовательность команд (кроме еще одной команды `MACRO`). Имена макро могут состоять только из маленьких латинских букв от 'a' до 'z' и иметь длину от 1 до 10.
- Команда `REPEAT`. Если вы напишете `REPEAT <macro_name> <value>`, то последовательность команд, ассоциированная с соответствующим макро (который должен быть объявлен выше), будет последовательно выполнена `<value>` раз. Если передать отрицательное количество повторений, макро не будет вызван ни разу.

Переведите код из обычного псевдокода в более понятный вам, без вложенностей. И кто знает, может быть именно вы под руководством Роберта Оппенгеймера совершите очередной прорыв в ядерной физике!

Формат входных данных

В первой строке ввода дано целое число n — количество строк в псевдокоде ($1 \leq n \leq 1000$). Следующие n строк содержат сам псевдокод.

Псевдокод строго отформатирован согласно грамматике из условия, включая пробелы и отступы. Изначально отступ равен 0. Каждый следующий вложенный цикл имеет отступ на 4 пробела больше, чем предыдущий. Каждый цикл содержит символ '{' в конце строки с объявлением и заканчивается строкой, содержащей только символ '}' на нужном отступе.

Имена переменных — строки из маленьких латинских букв от 'a' до 'z' и цифр от '0' до '9' длины от 1 до 10. Имя переменной не может быть равно `for`, `print` или `read` и не может начинаться с цифры. Все целочисленные константы неотрицательны и не превосходят 2000.

Если левая граница цикла больше правой, цикл совершает 0 итераций.

Гарантируется, что псевдокод не обращается к переменным, в которые до этого не было записано значение. Также гарантируется, что все используемые в объявлении цикла переменные имеют уникальные имена, не совпадающие с встречавшимися ранее переменными, используются только внутри соответствующего цикла, и не находятся слева от знака '=' или в аргументе операции `read`. **Никакая переменная в исходном псевдокоде не начинается на букву 'x'.**

Формат выходных данных

В первой строке выведите целое число $m \leq 5 \cdot n$ — количество строк в вашем переведенном коде. В следующих m строках выведите код в анти-вложенном псевдокоде в формате, описанном в условии.

Ваш код должен в точности повторять алгоритм, описанный псевдокодом из входных данных. **Не обязательно** добиваться того, чтобы все переменные имели те же имена и итоговые значения — достаточно, чтобы программы выводили одно и то же при условии одинакового ввода.

Все имена переменных в вашем коде должны состоять также из маленьких латинских букв и иметь длину от 1 до 10. Целочисленные константы в вашем коде должны быть неотрицательными и не превосходить 10000.

Примеры

стандартный ввод	стандартный вывод
<pre>6 n = 1 read(k) for i = 0...k { n = n + k } print(n)</pre>	<pre>9 n GETS 1 READ k MACRO increase: n GETS n + k iters GETS k + 1 REPEAT increase iters PRINT n</pre>
<pre>14 read(somevalue) read(morevalue) for i = 0...10 { for j = 1...somevalue { print(j) } smthidk = i wut = 42 for k = 1...morevalue { smthidk = smthidk + k wut = smthidk + smthidk } print(wut) }</pre>	<pre>23 MACRO printer: PRINT j j GETS j + 1 MACRO summator: smthidk GETS smthidk + k wut GETS smthidk + smthidk k GETS k + 1 MACRO outer: j GETS 1 REPEAT printer somevalue smthidk GETS i k GETS 1 wut GETS 42 REPEAT summator morevalue PRINT wut i GETS i + 1 READ somevalue READ morevalue i GETS 0 REPEAT outer 11</pre>

Задача J. Опасные опыты

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

Роберт Оппенгеймер спокойно сидел в своем кабинете, пока к нему не зашел один из его подопечных с запросом на определенное количество урана, подписанного разными исследовательскими группами. Известно, что всего в проекте участвуют n групп исследователей.

Быстро расписавшись и отослав подопечного, Роберт понял, что не посмотрел на то, сколько урана запросила каждая группа. Поскольку он — гениальный ученый, то, исходя из своих знаний и понимания сути проводимых опытов, Роберт быстро вычислил, что критическая масса урана в исследовании i -й группы равна a_i . Теперь он задумался, а сколько должно прибыть и быть распределено урана между группами, чтобы хотя бы у одной из исследовательских групп его количество **гарантированно** набрало критическую массу вне зависимости от того, сколько урана каждая группа запросила.

К несчастью, внезапно зашел Эрнест Лоуренс, коллега Оппенгеймера, и забрал Роберта на совещание. Но перед уходом Роберт протянул вам, своему стажеру, листок, на котором записаны все критические массы для каждой группы. Сможете в его отсутствие посчитать минимальную суммарную массу распределяемого урана, гарантирующую набор критической массы хотя бы в одной группе?

Формат входных данных

В первой строке ввода дано целое число n — количество исследовательских групп в проекте Манхеттен ($1 \leq n \leq 2 \cdot 10^5$).

В следующей строке через пробел перечислены n целых чисел a_i — значения критической массы в исследовании i -й группы ($0 \leq a_i \leq 10^9$). Да, критическая масса может быть равна нулю, если исследователям не нужен уран, чтобы устроить полный хаос.

Формат выходных данных

Выведите единственное число — минимальное количество урана для гарантированного набора критической массы на одном из опытов.

Примеры

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

Замечание

Критическая масса — в ядерной физике минимальная масса делящегося вещества, необходимая для начала самоподдерживающейся цепной реакции деления. Коэффициент размножения нейтронов в таком количестве вещества больше единицы или равен единице. Размеры, соответствующие критической массе, также называют критическими. При массе больше критической цепная реакция может (при соответствующих условиях) лавинообразно ускоряться, что приводит к ядерному взрыву.

Задача К. Идеальная пара

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

В мире Барби важно, чтобы любая пара Барби-Кен была идеальна, ведь нет ничего прекрасней идеального. Но не всем везет в поиске партнера, поэтому был создан «Клуб Поиска Идеального Партнера».

Для того что бы облегчить поиск партнера, каждый впервые пришедший должен был пройти опрос, а после, в зависимости от ответов, каждой Барби и каждому Кену выдавалась строка, означающая его тип. Она так же размещалась на специальной доске в здании клуба, где любой желающий мог подойти и присмотреть себе подходящего партнера.

К несчастью, никто не хотел переводить специальные строки в более понятные характеристики, поэтому партнеров стали выбирать так: брали строчку Барби и склеивали ее со строчкой Кена, и если получившаяся строчка оказывалась палиндромом, то пара считалась идеальной, ведь не существует строки идеальнее палиндрома.

В скором времени для того, чтобы собирать статистику по возможным идеальным парам, решили после каждого обновления доски выводить, сколько идеальных пар может быть создано из анкет, закрепленных на доске. Доска представляет из себя два поля: поле со строками Кенов и поле со строками Барби. Изначально на доске нет ни одной строки. И каждый раз, когда кто-то проходит тест, после его окончания результат закрепляется на доску. Также иногда кто-то из участников клуба может найти себе партнера (из клуба или снаружи), после чего его строку снимают с доски.

Для каждого изменения доски выведите, сколько идеальных пар можно составить из текущего набора строк.

Формат входных данных

В первой строке ввода дано целое число t — количество изменений доски ($1 \leq t \leq 10^6$).

Изменения задаются следующим образом:

- « $1 + b$ » — Барби пришла в клуб, и по результатам теста получила строку b .
- « $1 - b$ » — Барби со строкой b нашла себе партнера и ушла из клуба. Гарантируется, что такая строка находилась на доске.
- « $2 + k$ » — Кен пришел в клуб, и по результатам теста получил строку k .
- « $2 - k$ » — Кен со строкой k нашел себе партнера и ушел из клуба. Гарантируется, что такая строка находилась на доске.

Формат выходных данных

После каждого изменения доски выведите, сколько идеальных пар можно составить из текущих строк.

Пример

стандартный ввод	стандартный вывод
6	0
1 + ken	1
2 + nek	1
1 + barbie	2
2 + ibrab	1
1 - ken	0
1 - barbie	

Задача L. Беспорядки в Барбилэнде

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

Как известно, Барбилэнд — прекрасное место, в котором все идеально, и все занимаются тем, чем должны заниматься по задумке своих создателей, компании «Mattel». Вот только Кену Стереотипной Барби не хватает общения с ней, поэтому он в какой-то момент решил проанализировать граф знакомств между всеми жителями Барбилэнда, чтобы понять, почему ему так одиноко. Несмотря на то, что ему это не помогло, опишем несколько фактов, которые он выяснил.

Всего в Барбилэнде ровно n жителей, и среди них ровно m пар общающихся друг с другом. У каждого жителя v есть показатель *статуса в обществе* p_v , а у каждой пары общающихся жителей (u, v) — показатель *близости* $d_{u,v}$. Тогда *недовольство* жителя v вычисляется как

$$s_v = \sum_{(u,v)} p_v \oplus d_{u,v},$$

где за \oplus обозначена операция **xor** (побитовое исключающее «ИЛИ»). Общее недовольство всех жителей вычисляется как сумма значений недовольства по всем жителям Барбилэнда.

После нескольких перемещений Кена и Барби между Барбилэндом и реальным миром, как мы знаем, весь Барбилэнд был в хаосе: несколько смен режима, многие пары разругались, и это еще только вершина айсберга! Пока все не вернулось в более спокойное состояние, было решено во избежание новых проблем временно запретить общение некоторым парам жителей, то есть удалить ребра из графа знакомств. При этом граф знакомств должен остаться связным, чтобы общество не раскололось на несколько независимых групп.

Тут то и пригодится информация, собранная Кеном ранее! Определите, какого минимального суммарного недовольства жителей можно добиться, удалив некоторые ребра из графа знакомств (общения), чтобы при этом для любых двух жителей все еще существовала цепочка общения, приводящая от одного к другому.

Формат входных данных

В первой строке даны два целых числа n и m — количество жителей и общавшихся пар ($1 \leq n, m \leq 2 \cdot 10^5$).

Во второй строке через пробел перечислены n целых чисел p_i — статус в обществе каждого жителя ($0 \leq p_i \leq 10^9$).

В следующих m строках перечислены ребра графа, по одному на строке. Каждая строка содержит три целых числа v_i , u_i и d_{u_i,v_i} , означающих, что жители u_i и v_i общаются с близостью d_{u_i,v_i} ($1 \leq u_i, v_i \leq n$; $0 \leq d_{u_i,v_i} \leq 10^9$).

Гарантируется, что $u_i \neq v_i$ для всех i , и что граф связен. Кратные ребра не запрещаются.

Формат выходных данных

Выведите единственное целое число — минимальное возможное суммарное недовольство, которого можно добиться, удалив некоторые ребра графа, и при этом оставив его связным.

Примеры

стандартный ввод	стандартный вывод
4 5 1 1 4 1 1 2 2 1 3 2 1 4 3 2 3 5 3 4 2	15
3 3 1 4 16 1 2 17 2 3 17 1 3 17	39

Замечание

Во втором примере выгодно выбрать ребра $1 \leftrightarrow 3$ и $2 \leftrightarrow 3$. Тогда недовольство первого жителя равно $1 \oplus 17 = 16$, второго — $4 \oplus 17 = 21$, а третьего — дважды по $16 \oplus 17 = 1$, так как веса всех ребер равны. Сумма получается равна 39.