

Задача А. Дом, в котором живет кот

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

Хозяева кота Бенедикта живут в самом обыкновенном доме. В этом доме n этажей, и если он чем-то и примечателен — то тем, что на каждом этаже этого дома найдётся хотя бы одна квартира, в которой живёт хотя бы один кот или кошка.

Кот Бенедикт живёт на этаже b , и номер этого этажа вам нужно будет определить, зная следующие два факта.

Пусть u — количество всех котов и кошек, живущих на этажах, расположенных выше этажа b .

Пусть d — количество всех котов и кошек, живущих на этажах, расположенных ниже этажа b .

Первый факт: разность чисел u и d неотрицательна: $u - d \geq 0$.

Второй факт: разность чисел u и d минимально возможная среди всех неотрицательных.

Ваша задача — определить этаж, на котором живет Бенедикт.

Примечание. Бенедикт учитывается при подсчёте котов и кошек.

Формат входных данных

В первой строке содержится целое число n ($3 \leq n \leq 10^4$) — количество этажей в доме.

В каждой из следующих n строк содержится по одному целому числу a_i ($1 \leq a_i \leq 100$, $i = 1, 2, \dots, n$), a_i — количество кошек на этаже i .

Формат выходных данных

Выведите единственное целое число — номер этажа, на котором живет Бенедикт.

Система оценки

Баллы за подзадачи начисляются только в случае прохождения всех тестов этой подзадачи. Участнику сообщается либо номер первого непройденного теста и результат проверки на этом тесте, либо что все тесты подзадачи пройдены.

Для второй подзадачи требуется, чтобы программа верно решала первую подзадачу.

Подзадача	Баллы за подзадачу	Ограничения	Необходимые подзадачи	Информация о проверке
1	30	$n \leq 100$	нет	первая ошибка
2	70	$n \leq 10^4$	1	первая ошибка

Примеры

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

Замечание

Поясним первый из приведённых примеров.

В доме 4 этажа.

Предположим, что Бенедикт живёт на первом этаже. В этом случае количество кошек и котов, которые живут на этажах, расположенных ниже, равно 0. А вот количество кошек и котов, которые живут на этажах, расположенных выше, составит $3 + 2 + 4 = 9$. Таким образом, разность $u - d = 9 - 0 = 9$.

Теперь предположим, что Бенедикт живёт на втором этаже. В этом случае количество кошек и котов, которые живут на этажах, расположенных ниже, равно 1. Количество же кошек и котов, которые живут на этажах, расположенных выше, составит $2 + 4 = 6$. Соответственно, разность $u - d = 6 - 1 = 5$.

Предположим, что Бенедикт живёт на третьем этаже. Тогда количество кошек и котов, которые живут на этажах, расположенных ниже, равно $4 = 1 + 3$. Количество же кошек и котов, которые живут на этажах, расположенных выше, также составит 4. Разность $u - d = 4 - 4 = 0$.

Вообще говоря, уже можно сделать вывод, что Бенедикт живёт на третьем этаже, поскольку, согласно условию, $u - d \geq 0$, а значит, 0 — это минимально возможное значение, которое подходит.

Тем не менее, выполним проверку для предположения, что Бенедикт живёт на 4 этаже. В этом случае $d = 1 + 3 + 2 = 6$, а $u = 0$. Разность $u - d = -6 < 0$, и этот ответ нам точно не подойдёт.

Задача В. Коты на заборе

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

Кот Бенедикт часто устраивается поудобнее на каком-то из подоконников и наблюдает из окна за происходящим вокруг дома.

Из одного окна виден сад, огороженный высоким забором. Забор имеет форму прямоугольника со сторонами длиной a и b . Бенедикту видна только одна из сторон длиной a .

На расстоянии d от левого конца этой стороны к забору примыкает дерево, по которому на забор взбираются уличные коты.

Кот $\#j$ приходит к дереву в момент t_j , взбирается на забор (считайте, что он делает это мгновенно), после чего идёт по забору с единичной скоростью. Если скорость положительная, кот будет двигаться в направлении «по часовой стрелке», если же скорость отрицательная, кот будет двигаться в направлении «против часовой стрелки».

Каждый кот желает совершить полный обход забора и спуститься по дереву обратно. Однако если по дороге он встретится с другим котом, то они спрыгнут с забора, чтобы выяснить, кто из них так некстати оказался на дороге у другого. Будем считать, что после этого разбирательства коты расходятся по своим делам и на забор уже не возвращаются.

Бенедикт может наблюдать разбирательство между котами, если оно происходит в любой точке видимой им стороны забора. Точки соединения этой стороны забора с другими также считайте видимыми.

Известны моменты, в которые коты приходили к дереву. Ваша задача — определить, скольким из них удалось полностью обойти забор, а также сколько разбирательств между котами мог наблюдать Бенедикт.

Важное замечание. Если некоторый кот завершил обход забора и в этот момент встретился с другим котом (так же завершившим обход забора или, напротив, только начинающим обход), никакого разбирательства не случится.

Формат входных данных

В первой строке содержатся целые числа n, a, b, d ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq a, b \leq 5 \cdot 10^4$, $0 \leq d \leq a$) — количество котов, которые приходили к дереву, длины сторон забора и расстояние от левого края стороны забора до дерева.

В следующей строке содержатся целые числа t_1, t_2, \dots, t_n ($1 \leq |t_j| \leq 10^9$, $j = 1, 2, \dots, n$) — моменты времени, в которые коты приходили к дереву. Если $t_j < 0$, это означает, что кот пойдёт по забору в направлении «против часовой стрелки», если же $t_j > 0$, то кот пойдёт по забору в направлении «по часовой стрелке».

Гарантируется, что $|t_1| < |t_2| < \dots < |t_n|$.

Формат выходных данных

В первой строке выведите два целых числа — количество котов, которым удалось полностью обойти забор, а также количество разбирательств между котами, которое пронаблюдал Бенедикт.

Система оценки

Баллы за подзадачи начисляются только в случае прохождения всех тестов этой подзадачи. Участнику сообщается либо номер первого непройденного теста и результат проверки на этом тесте, либо что все тесты подзадачи пройдены.

Для третьей подзадачи требуется, чтобы программа верно решала первую и вторую подзадачи.

Подзадача	Баллы за подзадачу	Ограничения	Необходимые подзадачи	Информация о проверке
1	30	$n \leq 100, a, b \leq 100, t_j \leq 5 \cdot 10^4$	нет	первая ошибка
2	10	гарантируется, что $sign(t_j) \neq sign(t_{j+1})$	нет	первая ошибка
3	60	любые возможные значения	1	первая ошибка

Примечание. $sign(t_j)$ — знак числа t_j .

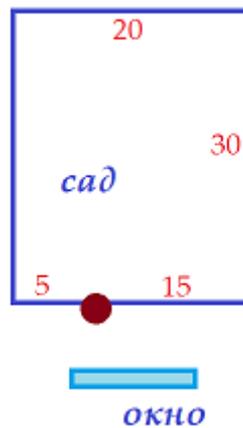
Пример

стандартный ввод	стандартный вывод
11 20 30 5	3
10 -25 -28 120 125 180 -200 -320	2
-390 420 -450	

Замечание

Данные о моментах, в которые коты приходили к дереву, в примере размещены в две строки для удобства печати. В тестах на сервере все данные об этих моментах содержатся в одной строке.

Поясним приведённый пример.



В момент 10 в обход по часовой стрелке отправится кот. В момент 25 в обход против часовой стрелки отправится другой кот. Эти коты встретятся на заборе, но Бенедикт их не увидит. Тем не менее, завершить обход им не удастся.

Далее в момент 28 очередной кот отправится в обход против часовой стрелки. В обход по часовой стрелке следующий кот отправится в момент 120. Эти коты встретятся в метре от левого конца стены забора, который виден Бенедикту. Так что это разбирательство между котами Бенедикт пронаблюдает. Разумеется, эти коты тоже не завершат обход.

В момент 125 следующий кот отправится в обход по часовой стрелке. А в момент 200 очередной кот отправится в обход против часовой стрелки. Между этими котами также случится разбирательство, которое увидит Бенедикт: за 2.5 метра от правого конца стены забора, которую Бенедикт может видеть.

Кот, который отправится в момент 180 в обход по часовой стрелке, благополучно доберётся до дерева, полностью обойдя забор, поскольку очередной кот, который отправится в обход против часовой стрелки, сделает это в момент 320.

Этот кот встретится с котом, который отправится в обход по часовой стрелке в момент 420. Но они встретятся на дереве, поэтому не мешают друг другу.

Однако кот, который отправится в обход в момент 420, встретит другого кота, который отправился в обход против часовой стрелки в момент 390. Впрочем, их разбирательство пройдёт вне поля зрения Бенедикта.

Кот же, который отправится в обход против часовой стрелки в момент 450, никого не встретит и завершит обход забора.

Таким образом, полный обход забора совершат три кота, а Бенедикт увидит разбирательства между двумя парами котов.

Задача С. Воробьи и лужа

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

Кот Бенедикт нежится на солнышке. Он расположился на подоконнике, так что ему хорошо видно происходящее внизу.

А внизу есть большая лужа, оставшаяся после ночного дождя. И к этой луже постоянно прилетают стайки воробьёв, желая утолить жажду. Каждая птица проводит возле лужи очень короткое время и сразу же улетает.

Известны минуты, в которые прилетали воробьи. Если воробей прилетел к луже в некоторую минуту, то в следующую минуту его уже нет возле лужи.

Бенедикт забрался на подоконник в **начале** какой-либо минуты (не ранее минуты 0). Спустился с подоконника Бенедикт в **конце** какой-либо минуты (не позднее последней минуты, когда прилетали воробьи).

Считайте, что воробьи прилетали после начала минуты и улетали до конца той же минуты.

Бенедикт подсчитал общее количество увиденных им воробьёв, получил число g и полагает, что если и ошибся в подсчётах, то не очень сильно.

Ваша задача — определить минимально возможное количество полных минут, которое Бенедикт провёл на подоконнике, полагая, что в течение этого времени он увидел количество воробьёв, которое отличается от g на минимально возможную величину.

Важное замечание. Совершенно точно известно, что Бенедикт видел по меньшей мере одного воробья.

Формат входных данных

В первой строке содержатся целые числа n и g ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq g \leq 10^{15}$) — количество различных минут, в которые прилетали воробьи, и количество воробьёв, подсчитанных Бенедиктом.

Во второй строке содержится n целых чисел t_1, t_2, \dots, t_n ($0 \leq t_1 < t_2 < \dots < t_n \leq 10^9$, $j = 1, 2, \dots, n$), t_j — минута, для которой есть запись о воробьях.

В третьей строке содержится n целых чисел v_1, v_2, \dots, v_n ($1 \leq v_j \leq 10^9$, $j = 1, 2, \dots, n$), v_j — количество воробьёв, которые прилетали в минуту t_j .

Формат выходных данных

В первой строке выведите целое число — минута, в начале которой Бенедикт взобрался на подоконник.

Во второй строке выведите целое число — минимально возможное количество полных минут, которое Бенедикт провёл на подоконнике.

В третьей строке выведите целое число — минимально возможное количество воробьёв, на которое Бенедикт ошибся в подсчётах.

Если количество подсчитанных Бенедиктом воробьёв меньше фактического, выведите число в третьей строке со знаком «-», в противном случае знак этого числа указывать не нужно.

Если возможно несколько вариантов ответа, выведите любой из них.

Система оценки

Баллы за подзадачи начисляются только в случае прохождения всех тестов этой подзадачи. Участнику сообщается либо номер первого непройденного теста и результат проверки на этом тесте, либо что все тесты подзадачи пройдены.

Для второй и третьей подзадач требуется, чтобы программа верно решала предыдущие подзадачи.

Подзадача	Баллы за подзадачу	Ограничения	Необходимые подзадачи	Информация о проверке
1	20	$n \leq 100, v_j \leq 10^6$	нет	первая ошибка
2	20	$n \leq 2000, v_j \leq 10^6$	1	первая ошибка
3	60	любые возможные значения	2	первая ошибка

Примеры

стандартный ввод	стандартный вывод
8 32 5 11 14 21 23 30 35 42 14 8 20 5 17 11 3 20	21 10 -1
8 17 5 11 14 21 23 30 35 42 14 8 12 5 17 11 3 20	23 1 0
8 35 5 11 14 21 23 30 35 42 14 8 12 7 18 6 3 20	5 10 1

Замечание

Поясним первый из приведённых примеров.

Бенедикт насчитал 32 воробья. Можно проверить, что наиболее близкими по количеству воробьёв оказываются два отрезка времени.

Если Бенедикт забрался на подоконник в минуту 11 (в начале минуты), он мог провести там 11 минут и спуститься в конце 21 минуты. В этом случае он увидел бы $33 = 8 + 20 + 5$ воробьёв, что на 1 отличается от результатов его подсчётов.

Если Бенедикт забрался на подоконник в минуту 21 (в начале минуты), он мог провести там 10 минут, и по окончании минуты 30 спуститься. В этом случае он увидел бы $33 = 5 + 17 + 11$ воробьёв, но одного из них не сосчитал.

Поскольку необходимо определить минимально возможное количество полных минут, выбираем второй отрезок времени.

Все остальные варианты выбора отрезков времени приводят к суммам, отличающимся от числа 32 на большую величину.

Задача D. Котофото

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

Хозяйка кота Бенедикта очень любит его фотографировать. Каждый день она делает несколько фотографий и помещает эти фотографии в папку *AllPhotos*. Конечно, у неё есть своя система именования фотографий, мы же для простоты будем полагать, что все фотографии занумерованы числами, начиная с 1, в том порядке, в котором они были сделаны.

Каждый вечер (после того, как она сделала все фотографии в этот день) она выбирает самую лучшую фотографию из всех, сделанных до этого дня включительно, и перемещает её в папку *TheBestPhotos*.

По прошествии n дней хозяйка кота Бенедикта решила отправить n самых лучших фотографий на фотоконкурс. Однако она поняла, что, если ей хочется выбрать n самых лучших фотографий, сделанных в течение n дней, ей придётся просмотреть не только папку *TheBestPhotos*, но ещё ряд других фотографий из папки *AllPhotos*.

Вам известно, какие фотографии и в какой последовательности были помещены в папку *TheBestPhotos* в течение n дней.

Ваша задача — определить, сколько фотографий из папки *AllPhotos* придётся просмотреть хозяйке кота Бенедикта, чтобы выбрать m лучших.

Формат входных данных

В первой строке содержатся целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество дней, в течение которых были сделаны фотографии.

Во второй строке содержится n целых чисел d_1, d_2, \dots, d_n ($1 \leq d_i \leq 10^4$, $i = 1, 2, \dots, n$), d_i — количество фотографий, сделанных в день $\#i$.

В третьей строке содержится n целых чисел f_1, f_2, \dots, f_n ($1 \leq f_i \leq \sum_{q=1}^i d_q$, $i = 1, 2, \dots, n$), f_i — номер фотографии, которую в день $\#i$ переместили в папку *TheBestPhotos*.

Гарантируется, что для любых $i \neq j$ $f_i \neq f_j$.

Формат выходных данных

Выведите целое число k — минимальное количество фотографий из папки *AllPhotos*, которое придётся просмотреть хозяйке кота Бенедикта (дополнительно к содержимому папки *TheBestPhotos*), чтобы выбрать n лучших.

Система оценки

Баллы за подзадачи начисляются только в случае прохождения всех тестов этой подзадачи. Участнику сообщается либо номер первого непройденного теста и результат проверки на этом тесте, либо что все тесты подзадачи пройдены.

Для второй и третьей подзадач требуется, чтобы программа верно решала предыдущие подзадачи.

Подзадача	Баллы за подзадачу	Ограничения	Необходимые подзадачи	Информация о проверке
1	15	$n \leq 20, d \leq 100$	нет	первая ошибка
2	25	$n \leq 100, d \leq 10^4$	1	первая ошибка
3	60	любые возможные значения	2	первая ошибка

Примеры

стандартный ввод	стандартный вывод
5 3 4 2 3 2 3 5 8 7 4	7
5 3 4 2 3 2 3 5 2 7 4	0

Замечание

Поясним первый из приведённых примеров.

Для удобства запишем диапазоны номеров фотографий, сделанных в тот или иной день.

В первый день были сделаны фотографии с номерами 1 — 3.

Во второй день были сделаны фотографии с номерами 4 — 7.

В третий день были сделаны фотографии с номерами 8 — 9.

В четвёртый день были сделаны фотографии с номерами 10 — 12.

В пятый день были сделаны фотографии с номерами 13 — 14.

В первый день была выбрана фотография #3, сделанная в первый день.

Во второй день была выбрана фотография #5, сделанная во второй день. При этом мы не можем быть уверены, что остальные фотографии не лучше фотографии #3: на момент её выбора они ещё не были сделаны.

В третий день была выбрана фотография #8, сделанная в этот же день, и мы можем быть уверены, что она лучше фотографии #9, а также всех, не выбранных ранее. А вот фотография #9 может быть лучше фотографии #3 (или не лучше, мы этого не знаем).

В четвёртый день была выбрана фотография #7. Это позволяет нам заключить, что фотографии с #10 по #12 могут быть лучше разве что фотографии #3 среди выбранных ранее.

Наконец, в пятый день была выбрана фотография #4. И это тоже оставляет неопределённость только в отношении фотографии #3.

Таким образом, нужно просмотреть 7 фотографий, которые потенциально могут заменить фотографию #3: это фотографии 6, 9, 10, 11, 12, 13, 14.

Задача Е. Вкусный план

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

Кот Бенедикт совершенно не против полезного корма, если, конечно, он вкусный. Хозяйева Бенедикта долго подбирали корм, который нравился бы не только им (с точки зрения полезности и, конечно, стоимости), но и коту. Поэтому когда такой корм появляется в продаже по цене со скидкой, стараются приобрести его впрок.

Однако есть определённые ограничения. Во-первых, хозяин кота Бенедикта не хочет хранить в квартире более m упаковок корма. Это означает, что ни в какой момент в квартире не должно находиться одновременно более m упаковок корма (хозяин пересчитает их, прежде чем Бенедикт сможет съесть одну). Считайте, что корм всегда покупается в первой половине дня, а выдаётся коту во второй половине дня.

Во-вторых, у корма есть срок годности, и хозяйка Бенедикта считает недопустимым кормить кота кормом с истекшим сроком годности.

Известны n дней, в которые этот корм будет продаваться со скидкой. Для каждого из этих дней также известно, в течение скольких дней будет годен корм, который в этот день продаётся.

Считайте, что в первый день запасы корма отсутствуют, иными словами, составляют 0 упаковок.

Если корм приобретается в некоторый день, то кот может съесть его в этот же день.

Если в некоторый день у хозяев имеется в запасе хотя бы одна упаковка корма (конечно же, с не истекшим ещё сроком годности), то кот получит этот корм.

Если в некоторый день у хозяев не окажется корма в запасе, и в этот день они не могут приобрести корм со скидкой, они купят ровно одну упаковку корма, которую кот сразу же съест.

Определим номер дня D как максимальный номер дня, в который может истечь срок годности корма, закупленного в один из n дней.

Ваша задача — составить план закупок корма таким образом, чтобы минимизировать количество дней со дня 1 до дня D включительно, в которые хозяйевам придётся приобретать корм без скидки.

При составлении плана закупок следует помнить, что хозяйка Бенедикта будет очень расстроена, если хотя бы одна упаковка корма не будет использована по причине окончания срока годности. Поэтому все купленные упаковки обязательно должны быть израсходованы.

Примечание. Если в некоторый день корм будет годен в течение 0 дней, то он должен быть съеден котом в этот день.

Формат входных данных

В первой строке содержатся целые числа n и m ($1 \leq n \leq 2 \dots 10^5$, $1 \leq m \leq 10^6$) — количество дней, в которые корм будет продаваться со скидкой, и максимально возможное количество упаковок, которые хозяин кота Бенедикта соглашается хранить в квартире.

Во второй строке содержатся целые числа d_1, d_2, \dots, d_n ($1 \leq d_1 < d_2 < \dots < d_n \leq 10^9$) — номера дней, в которые корм будет продаваться со скидкой.

В третьей строке содержатся целые числа g_1, g_2, \dots, g_n ($0 \leq g_i \leq 10^6$, $i = 1, 2, \dots, n$), g_i — количество дней со дня покупки, в течение которых будет годен корм, приобретаемый в день $\#i$.

Формат выходных данных

В первой строке выведите минимально возможное количество дней до дня D включительно, в которые хозяйевам придётся приобретать корм без скидки.

Во второй строке выведите n чисел. На позиции $\#i$ укажите количество упаковок корма, которое следует приобрести хозяйевам в день d_i .

Если существует несколько вариантов ответа, выведите любой из них.

Система оценки

Баллы за подзадачи начисляются только в случае прохождения всех тестов этой подзадачи. Участнику сообщается либо номер первого непройденного теста и результат проверки на этом тесте, либо что все тесты подзадачи пройдены.

Для третьей подзадачи требуется, чтобы программа верно решала предыдущие подзадачи.

Подзадача	Баллы за подзадачу	Ограничения	Необходимые подзадачи	Информация о проверке
1	20	$n, m \leq 2 \cdot 10^3$ $d_n \leq 10^6$	нет	первая ошибка
2	20	$g_1 = g_2 = \dots = g_n$	нет	первая ошибка
3	60	любые возможные значения	2	первая ошибка

Пример

стандартный ввод	стандартный вывод
8 10 2 14 19 23 28 35 44 51 20 7 30 10 4 3 11 2	5 10 5 10 4 5 4 9 3

Замечание

Поясним приведённый пример.

В день 1 корм будет куплен без скидки.

В день 2 будет приобретено 10 упаковок корма (больше хозяин не соглашается хранить дома, хотя срок годности у этого корма позволяет закупить его в большем количестве).

Следующий день, когда будут скидки, — день 14. Так что в дни 12 и 13 придётся покупать корм без скидки.

В день 14 срок годности у корма будет достаточно коротким — 7 дней, в отличие от корма, который будет продаваться в день 19. Поэтому достаточно купить 5 упаковок такого корма, чтобы его хватило ровно до дня 18 включительно.

В день 19 можно приобрести 10 упаковок корма: срок годности позволяет и больше, но хозяин против.

К дню 23 кот съест 4 упаковки корма, приобретённого в день 19. Так что запасы можно пополнить, докупив 4 упаковки, теперь уже со сроком годности до дня 33.

В день 28 из запасов в наличии будут 5 упаковок корма, купленного ранее. Заметим, что может оказаться разумным сначала израсходовать 4 упаковки со сроком годности до дня 33, а затем уже отдать коту одну из упаковок со сроком годности до дня 49 (приобретённую в день 19).

Приобрести в день 28 можно 5 упаковок: кот может сразу приступить к их поеданию. В день 32 он съест все эти упаковки, после чего в дни 33 и 34 хозяева отдадут ему ещё две упаковки, приобретённые в день 19.

В день 35 в распоряжении хозяев останется 3 упаковки, купленных в день 19. Однако, поскольку корм, который будет продаваться в день 35, имеет срок годности всего 3 дня, они ограничатся приобретением 4 упаковок, которые сразу же будут выдаваться коту. В дни с 39 по 41 кот съест те самые три упаковки, купленные в день 19. А в дни 42 и 43 хозяевам придётся приобрести корм без скидки.

В день 44 корм снова будет продаваться со скидкой. Срок годности этого корма — 11 дней, и именно он определит день D : план закупок будет составляться до дня 55 включительно.

В этот день можно приобрести 10 упаковок корма; в предлагаемом решении указано 9 упаковок, на окончательный ответ выбор того или иного варианта не повлияет (повлияет лишь на количество корма, который будет приобретён в день 51). В случае, когда приобретается 9 упаковок, кот съест 7 из них до дня 50 включительно.

В день 51 корм снова будет продаваться со скидкой, но приобрести больше 3 упаковок не получится: он годен лишь до дня 53 включительно. Так что до дня 53 кот съест именно эти упаковки, а в дни 54 и 55 съест оставшиеся 2 упаковки, приобретённые в день 44.

Таким образом, хозяевам придётся покупать корм без скидки в дни 1, 12, 13, 42 и 43 — итого 5 дней. Составить более удачный план закупок не получится.

Задача F. Большая коробка

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

Кот Бенедикт очень любит забираться в коробки. Недавно он заметил, как его хозяйева убирали в кладовку большую коробку. Улучив момент, он проник в кладовку и обнаружил, что коробку поставили на одну из верхних полок. Бенедикт смог забраться на полку выше коробки, но, когда он попытался попасть внутрь коробки, его постигла неудача. Коробка и некоторые другие вещи начали падать, и коту пришлось в спешном порядке ретироваться.

Коробка представляла собой прямоугольный параллелепипед ненулевого объема с целочисленными координатами вершин. После вмешательства Бенедикта ровно K вершин параллелепипеда изменили координаты.

Ваша задача — восстановить координаты вершин параллелепипеда. Если возможных вариантов ответа несколько — выведите любой из них.

Формат входных данных

В первой строке содержится целое число k ($1 \leq k \leq 3$) — количество вершин, координаты которых изменились.

Следующие 8 строк содержат по три целых числа через пробел $x_i \ y_i \ z_i$ ($-23 \cdot 10^6 \leq x_i, y_i, z_i \leq 23 \cdot 10^6$) — координаты вершин коробки после изменения.

Гарантируется, что существует хотя бы один прямоугольный параллелепипед ненулевого объема, такой, что не менее $(8 - k)$ введенных точек совпадают с его различными вершинами.

Формат выходных данных

Выведите 8 строк, содержащих по три целых числа через пробел $x_i \ y_i \ z_i$ — изначальные координаты вершин коробки (параллелепипеда).

Способ восстановления будет считаться корректным, если **ровно** k выведенных вершин параллелепипеда не присутствовали во входных данных.

Если корректных способов восстановить координаты вершин параллелепипеда несколько, выведите любой из возможных вариантов.

Вершины можно выводить в любом порядке (порядок координат должен остаться $x \ y \ z$).

Система оценки

Баллы за подзадачи начисляются только в случае прохождения всех тестов этой подзадачи. Участнику сообщается либо номер первого непройденного теста и результат проверки на этом тесте, либо что все тесты подзадачи пройдены.

Для второй и третьей подзадач требуется, чтобы программа верно решала предыдущие подзадачи.

Подзадача	Баллы за подзадачу	Ограничения	Необходимые подзадачи	Информация о проверке
1	10	$k = 1$, стороны параллелепипеда параллельны осям координат	нет	первая ошибка
2	25	$2 \leq K \leq 3$, стороны параллелепипеда параллельны осям координат	1	первая ошибка
3	20	$K = 1$	1	первая ошибка
4	45	$2 \leq K \leq 3$	2, 3	первая ошибка

Пример

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

Замечание

Определение: прямоугольный параллелепипед — это многогранник, у которого 6 граней, причем все грани —прямоугольники.

Поясним первый приведённый пример.

Коробка представляла собой прямоугольный параллелепипед со сторонами, параллельными осям координат и следующими вершинами:

- $(-1, -1, -1)$;
- $(-1, 2, -1)$;
- $(3, -1, -1)$;
- $(3, 2, -1)$;
- $(-1, -1, 4)$;
- $(-1, 2, 4)$;
- $(3, -1, 4)$;
- $(3, 2, 4)$.

Таким образом, параллелепипед расположен от -1 до 3 по оси OX , от -1 до 2 по оси OY , от -1 до 4 по оси OZ .

В результате действий кота изменились две вершины:

- $(-1, 2, 4)$ на $(0, 0, 0)$;
- $(3, 2, -1)$ на $(3, -1, 4)$.

Задача G. Очень большая коробка

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

Когда кот Бенедикт изучал содержимое кладовки, он обратил внимание на стопку квадратных листов из толстого картона. По его мнению, эти листы, без сомнения, были получены при разрезании очень большой коробки.

Бенедикт полагает, что очень большая коробка имела форму прямоугольного параллелепипеда, но без верхней грани. Он насчитал q листов и теперь размышляет, какой максимально возможный объем мог бы быть у очень большой коробки, которую можно сложить из всех или некоторых имеющихся листов.

Ваша задача — определить этот максимально возможный объём.

Считайте, что все квадратные листы имеют сторону 1.

Формат входных данных

В первой строке содержится натуральное число q ($5 \leq q \leq 2023 \cdot 10^6$ — количество квадратных листов).

Формат выходных данных

Выведите натуральное число — максимально возможный объём очень большой коробки в виде прямоугольного параллелепипеда без верхней грани.

Система оценки

Баллы за подзадачи начисляются только в случае прохождения всех тестов этой подзадачи. Участнику сообщается либо номер первого непройденного теста и результат проверки на этом тесте, либо что все тесты подзадачи пройдены.

Для второй и третьей подзадач требуется, чтобы программа верно решала предыдущие подзадачи.

Подзадача	Баллы за подзадачу	Ограничения	Необходимые подзадачи	Информация о проверке
1	20	$q \leq 1000$	нет	первая ошибка
2	30	$q \leq 10^6$	1	первая ошибка
3	50	любые возможные значения	1,2	первая ошибка

Пример

стандартный ввод	стандартный вывод
55	36

Замечание

Поясним приведённый пример. Можно собрать такую коробку: дно размером 3×4 , высота — 3, использовано всего 54 квадратных листа.

Задача Н. Равномерное распределение

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

Кот Бенедикт считает важным хороший отдых. В квартире имеется m мест, на каждом из которых ему нравится лежать. Кот Бенедикт полагает, что хорошо отдохнул, если ему удалось полежать на месте $\#j$ в течение хотя бы t_j единиц времени.

Бенедикт не уверен, что его хозяева разделяют его воззрения относительно его любимых мест для лежания, поэтому (на всякий случай) устраивается полежать на месте $\#j$ только, если уверен, что хозяева будут отсутствовать в квартире в течение как минимум t_j единиц времени.

Для каждого из ближайших n дней Бенедикту известно, в течение какого времени хозяева будут отсутствовать в квартире.

В каждый из дней кот планирует лежать на каком-либо одном месте.

Кот хотел бы составить график лежания таким образом, чтобы минимальное количество раз, которое он лежал на каком-либо месте, отличалось от максимального количества раз, которое он лежал на каком-либо месте, на минимально возможную величину.

Ваша задача — определить эту минимально возможную величину, а также составить график лежания для кота, указав, на каком месте в какой день ему следует отдыхать.

Формат входных данных

В первой строке содержатся целые числа m и n ($2 \leq m, n \leq 10^5$) — количество любимых мест отдыха кота и количество ближайших дней, для которых нужно составить график.

Во второй строке содержится m целых чисел t_1, t_2, \dots, t_m ($1 \leq t_j \leq 10^6$, $j = 1, 2, \dots, m$), t_j — минимальное время, в течение которого кот хотел бы отдыхать на месте $\#j$.

В третьей строке содержится n целых чисел d_1, d_2, \dots, d_n ($\min(t_1, t_2, \dots, t_m) \leq d_i \leq 10^6$, $i = 1, 2, \dots, n$), d_i — время, в течение которого хозяева кота будут отсутствовать в квартире в день $\#i$.

Формат выходных данных

В первой строке выведите целое число — минимально возможную разность между максимальным количеством раз, которое кот будет лежать на каком-либо месте, и минимальным количеством раз, которое кот будет лежать на каком-либо месте.

Во второй строке выведите n целых чисел. На позиции $\#i$ должен быть записан номер места, на котором коту следует отдыхать в день $\#i$. Считайте, что места занумерованы с 1 в порядке упоминания их во входных данных.

Разделяйте числа пробелами.

Если существует несколько вариантов ответа, выведите любой из них.

Система оценки

Баллы за подзадачи начисляются только в случае прохождения всех тестов этой подзадачи. Участнику сообщается либо номер первого непройденного теста и результат проверки на этом тесте, либо что все тесты подзадачи пройдены.

Для третьей подзадачи требуется, чтобы программа верно решала предыдущие подзадачи.

Подзадача	Баллы за подзадачу	Ограничения	Необходимые подзадачи	Информация о проверке
1	20	$m = 2; n \leq 2000$	нет	первая ошибка
2	20	$n, m \leq 2000$	1	первая ошибка
3	60	любые возможные значения	2	первая ошибка

Примеры

стандартный ввод	стандартный вывод
5 14 3 11 8 7 7 5 9 12 7 8 10 10 7 8 6 14 18 20 10	1 1 4 3 1 5 5 3 5 4 1 2 2 2 3
3 7 12 15 10 11 10 14 10 12 10 13	4 3 3 1 3 1 3 1

Замечание

Поясним приведённые примеры.

В первом примере возможны и другие варианты ответа.

Можно заметить, что даже для места с самым длительным временем лежания существует 3 дня, в которые кот может на этом месте спокойно отдохнуть.

Однако поскольку любимых мест у кота 5, а план нужно составить на 14 дней, на каком-то из мест ему придётся лежать меньшее количество дней, чем на остальных.

Во втором примере не найдётся ни одного дня из 7, в который кот мог бы полежать на месте #2. Поэтому лучшее, что можно сделать — по возможности поровну распределить дни для оставшихся двух мест. Конечно, поскольку 7 не делится нацело на 2, для одного из мест дней, в которые кот будет на нём отдыхать, окажется как минимум на 1 больше. Можно удостовериться, что на месте 1 кот сможет отдохнуть 3 раза, а на месте 3 — 4 раза. Увеличить количество дней отдыха на месте 1 невозможно, а увеличение количества дней отдыха на месте 3 ухудшит итоговый результат.

Минимальное же количество раз, которое кот будет лежать на месте 2, составляет 0, так что минимально возможная разность составит $4 - 0 = 4$.

Задача I. Мячики

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

Кот Бенедикт иногда любит гонять мячик по квартире. Случается, что в процессе игры мячик закатывается под шкаф, из-под которого кот вытащить его не может. Это очень огорчает кота.

Чтобы Бенедикт меньше расстраивался, хозяйка приобрела ему ещё пару мячиков. Так что теперь у Бенедикта есть три мячика: красный, жёлтый и зелёный. Если какой-либо мячик закатывается под шкаф, кот может играть с другим мячиком. Периодически хозяйка кота Бенедикта достаёт мячики из-под шкафа. При этом она достаёт **все** мячики, которые оказались к этому моменту под шкафом: иногда один, иногда два, а иногда и все три.

Пронаблюдав за котом, хозяйка предположила, что у Бенедикта есть определённые предпочтения. Когда в его распоряжении оказываются три мячика, он всегда выбирает для игры мячик определённого цвета (мы не будем раскрывать, какого именно, в этом и состоит задача). Когда этот мячик оказывается под шкафом, Бенедикт выбирает из двух оставшихся также вполне определённый мячик. Третьим же мячиком он играет лишь тогда, когда недоступны два других.

В некоторый момент хозяйка Бенедикта начала записывать, с каким мячиком играл Бенедикт. Известна последовательность символов, описывающая, в каком порядке Бенедикт играл с мячиками. Символом R обозначен красный мячик, символом Y — жёлтый, символом G — зелёный мячик.

Какие мячики и в какие моменты находились под шкафом и когда их доставала хозяйка кота — не известно. Также неизвестно, сколько мячиков было в распоряжении Бенедикта, когда его хозяйка начала вести записи.

Ваша задача — определить, какой мячик нравится Бенедикту больше других и какой — меньше других.

Формат входных данных

В первой строке содержится непустая последовательность символов R , Y , G . Длина последовательности не превышает $2 \cdot 10^5$ символов.

Формат выходных данных

Выведите строку, состоящую из трёх символов.

Первый символ должен соответствовать цвету мячика, который более всего нравится Бенедикту. Последний символ должен соответствовать цвету мячика, который менее всего нравится Бенедикту. Второй символ должен соответствовать цвету оставшегося мячика.

Если предпочтения Бенедикта нельзя установить однозначно, выведите в качестве ответа строку NNN .

Если же хозяйка ошибается и у Бенедикта нет предпочтений, выведите строку FFF .

Система оценки

Баллы за подзадачи начисляются только в случае прохождения всех тестов этой подзадачи. Участнику сообщается либо номер первого непройденного теста и результат проверки на этом тесте, либо что все тесты подзадачи пройдены.

Для третьей подзадачи требуется, чтобы программа верно решала предыдущие подзадачи.

Подзадача	Баллы за подзадачу	Ограничения	Необходимые подзадачи	Информация о проверке
1	30	$n \leq 2000$	нет	первая ошибка
2	70	любые возможные значения	1	первая ошибка

Примеры

стандартный ввод	стандартный вывод
RYRRRYGGRRYYYYYGRY	RYG
GYRRRG	NNN
GYYYRRYYG	FFF

Замечание

Поясним приведённые примеры.

Если предположить, что в первом примере предпочтения Бенедикта таковы: красный мячик (R), жёлтый мячик (Y), зелёный мячик (G), можно убедиться, что заданная последовательность символов удовлетворяет этим предпочтениям. Также можно убедиться, что никакие иные предпочтения не дают возможности считать эту последовательность корректной.

Во втором примере можно подобрать более одного набора предпочтений, при котором эта последовательность окажется корректной. Например, можно считать, что Бенедикт предпочитает всем мячикам зелёный, в отсутствие зелёного играет жёлтым, в отсутствие жёлтого и зелёного — красным. Однако также можно предположить, что Бенедикт предпочитает всем мячикам жёлтый, в отсутствие жёлтого — красный, а зелёным будет играть, если два другие мячика недоступны. С зелёного же последовательность началась лишь потому, что, когда хозяйка Бенедикта начала вести записи, жёлтый и красный мячики находились под шкафом.

Подобрать же набор предпочтений для третьего примера не удаётся.