Задача A. Fence Painting

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Том и Гек красят забор. Забор состоит из $a + b$ сегментов, при чем первые a сегментов — деревянные, а оставшиеся b — металлические.

Тому и Геку обещали вознаграждение за покраску забора. На деревянные и металлические сегменты тратится разное количество усилий, поэтому покраска деревянного сегмента оценивается в x монет, а металлического — в y монет.

Том начинает красить забор с первого (крайнего левого) сегмента и красит подряд сегмент за сегментом. Гек, аналогично, начинает красить забор справа и двигается навстречу Тому. Разумеется, ребята хотят, чтобы вознаграждение за покраску забора было разделено справедливо: суммы, которые получат за забор Том и Гек, должны отличаться на минимальную возможную величину.

Помогите им определить, какое должно быть их место встречи k, чтобы вознаграждение за покраску Тому и Геку различалось как можно меньше. Местом встречи будем считать номер последнего сегмента, который покрасит Том. Иными словами, если место встречи — k, Том покрасил k левых сегментов, а Гек — оставшиеся $a + b - k$ правых.

Формат входных данных

В первой строке ввода даны два целых числа a и b — количество деревянных и металлических сегментов забора, соответственно ($1 \leqslant a, b \leqslant 10^9$).

Во второй строке ввода даны два целых числа x и y — количество монет, которое платят за покраску одного деревянного или металлического сегмента, соответственно ($1 \leqslant x, y \leqslant 10^9$).

Формат выходных данных

Выведите единственное целое число k — место встречи, при котором Том и Гек получат наиболее близкие по значению вознаграждения. Если оптимальных вариантов ответа несколько, выведите любой из них.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	**Баллы**	**Ограничения**	**Необходимые подзадачи**
0	–	примеры из условия	
1	16	$a, b \leqslant 1000$	0
2	19	$x = y$	
3	21	$a = b$	
4	24	$a, b \leqslant 10^6$	0, 1
5	20	нет	0 – 4

Примеры

стандартный ввод	стандартный вывод
1 1 1 1	1
5 10 2 1	5

Задача B. Work, Sleep, Repeat

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

В компании VK работают очень талантливые программисты. Поскольку им постоянно приходится решать интересные задачи, работа проходит быстро и с удовольствием. Однако это не значит, что при этом можно совсем от нее не отдыхать.

Программист Лев, (почти) занимающий позицию тимлида, решил взять себе отпуск. Но поскольку его присутствие на рабочем месте и доступность могут быть критичными для команды, он разобьет этот отпуск на несколько частей: а именно, он будет на протяжении обозримого будущего жить в цикличном режиме «x дней работы, затем y дней отпуска».

Разумеется, у него есть запланированные совещания с командой, дни релиза новой версии и другие события, которые никак нельзя пропускать. Известно всего n таких событий, и i-е из них запланировано на день d_i.

Считая, что сегодня день номер 1, а все события запланированы не раньше, чем на день номер $x+y+1$ определите, существует ли такой день $D \leqslant x+y$, что если в день D Лев перейдет на новый режим, он не пропустит ни одно запланированное событие. Иными словами, требуется добиться того, чтобы все дни d_i приходились на фазу работы в соответствующем цикле.

Формат входных данных

В первой строке ввода через пробел даны три целых числа n, x и y — количество запланированных событий, а также длина фазы работы и длина фазы отпуска, запланированных Львом ($1 \leqslant n \leqslant 2 \cdot 10^5$; $1 \leqslant x, y \leqslant 10^9$).

Во второй строке через пробел перечислены n целых чисел d_i — дни, на которые запланированы события ($x + y < d_i \leqslant 10^9$). **Не** гарантируется, что все d_i различны.

Формат выходных данных

Выведите единственное целое число от 1 до $x + y$ — номер дня, в который Льву следует перейти на новый режим. Если возможных ответов несколько, выведите любой из них. Если ответа не существует, выведите `-1`.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
0	–	примеры из условия	
1	15	$x = 1$	
2	19	$n, x, y \leqslant 1000$	0
3	18	$y = 1$	
4	21	$x \leqslant y$	0, 2
5	27	нет	0 – 4

Примеры

стандартный ввод	стандартный вывод
3 3 1 5 8 11	3
4 3 1 5 8 11 14	-1
5 4 3 10 16 17 18 99	1

Замечание

В первом примере если цикл начинается с третьего дня, Лев будет работать в дни 3 – 5, затем 7, 8, 9 и, наконец, 11 – 13.

Задача C. Palindromization

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

Палиндромы — последовательности, читающиеся одинаково как слева направо, так и справа налево. Такие последовательности играют большую роль в строковых алгоритмах. Программист Влад считает, что работать с палиндромами проще, чем с любыми другими строками и массивами, поэтому каждый раз, когда Влад видит массив, он стремится сделать из него палиндром.

В этот раз Влад получил массив a из n целых чисел. За одну операцию Влад может выбрать любой отрезок этого массива и прибавить ко всем элементам на этом отрезке одинаковое число x от 1 до k включительно.

Несмотря на то, что Влад привык работать с палиндромами, в этот раз у него возникли проблемы с тем, чтобы получить из a палиндром за минимальное число действий. Помогите ему поскорее справиться с этой задачей.

Формат входных данных

Первая строка ввода содержит два целых числа n и k — размер исходного массива и ограничение на прибавляемые значения ($1 \leqslant n \leqslant 10^6$; $1 \leqslant k \leqslant 3$).

Во второй строке ввода через пробел даны n целых чисел a_i — элементы массива ($1 \leqslant a_i \leqslant 10^9$).

Формат выходных данных

Выведите единственное число — минимальное количество операций, необходимое, чтобы сделать из массива a палиндром.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	**Баллы**	**Ограничения**	**Необходимые подзадачи**
0	–	примеры из условия	
1	10	$n, a_i \leqslant 10$	0
2	11	$a_i \leqslant 2$, $k = 1$	
3	12	$a_i \leqslant a_{i+1}$ для всех $i < n$	
4	20	$k = 1$	2
5	22	$k \leqslant 2$	4
6	23	$k \leqslant 3$, $n \leqslant 1000$	0, 1

Обратите внимание, что нет группы с $k = 3$ без каких-либо других дополнительных ограничений!

Примеры

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

Замечание

В первом примере нужно прибавить 1 к отрезку $[1, 3]$, прибавить 1 к отрезку $[1, 2]$, а затем два раза прибавить 1 к первому элементу. Таким образом после четырех операций массив равен $[5, 4, 4, 4, 5]$.

Во втором примере нужно прибавить 2 к отрезку $[1, 1]$, прибавить 2 к отрезку $[4, 5]$ и прибавить 1 к отрезку $[4, 5]$, после чего прибавить 1 к отрезку $[5, 5]$. Таким образом после четырех операций массив равен $[5, 6, 4, 4, 6, 5]$.

В третьем примере нужно прибавить 1 к отрезку $[1, 4]$, прибавить 3 к отрезку $[6, 7]$ и прибавить 1 к отрезку $[7, 7]$. Таким образом после трех операций массив равен $[2, 5, 4, 2, 2, 4, 5, 2]$.

Задача D. Hackathon

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

VK очень часто проводит хакатоны, где программисты решают разные задачи. Сейчас VK готовит новый хакатон, в котором все желающие могут попробовать себя в разработке уникального полезного приложения или сервиса для одного из нескольких треков, в числе которых Финтех, Культура, Медиа и другие.

Как и на любом хорошо организованном хакатоне, чтобы участники могли полностью посвятить себя решению поставленной задачи, им предоставляется тщательно организованное место проведения. В число приятных бонусов часто входят еда и напитки, которыми можно подкрепиться в течение нескольких дней, проводимых участниками на площадке.

В этот раз VK организует новый хакатон, и на него на площадке проведения собрались m участников. Для этих m участников организаторы заказали n пицц, которые на площадке расположили в ряд и пронумеровали от 1 до n. Известно, что

- каждая пицца состоит из бесконечного числа кусочков;
- первый кусочек i-й пиццы имеет толщину теста a_i, а каждый следующий — на 1 больше предыдущего (то есть $a_i + 1$, $a_i + 2$, и так далее);
- наиболее привлекательными для участников кажутся кусочки с наименьшей толщиной теста;
- любой участник может подойти к ряду с пиццами в произвольном месте, но после может перемещаться только влево (в сторону меньших номеров), чтобы не мешать другими участникам.

Периодически участники встают со своих мест и подходят к ряду пицц, чтобы взять один кусочек. Участник номер i подходит к пицце с номером s_i в момент времени t_i, после чего моментально находит кусочек пиццы с наименьшей толщиной теста среди пицц от 1-й до s_i-й и начинает двигаться в его сторону. На перемещение между двумя соседними пиццами у каждого участника уходит одинаковое время — v секунд. Несколько участников могут одновременно находиться около одной пиццы.

Как только участник подходит к пицце, кусочек в которой он выбрал, он его забирает и уходит обратно на свое место. Если несколько участников выбрали один и тот же кусочек и подошли к нему одновременно, кусочек забирает участник с меньшим номером. При этом если кусочек, на который рассчитывал участник, кто-то забрал, этот участник моментально выбирает новую цель по тому же алгоритму: он смотрит на все пиццы от первой до его текущей позиции, выбирает среди них минимум, и начинает двигаться к нему.

Формально этот процесс можно описать следующим образом. В каждое целое число секунд t:

1. все участники, уже стоявшие в ряду в момент времени $t - 1$, сдвигаются на $\frac{1}{v}$ влево;
2. все участники с $t_i = t$ подходят к пицце с номером s_i;
3. время «останавливается»;
4. определяются все участники, стоящие на таких **целых** позициях p, что $a_p = \min(a_1, a_2, \ldots, a_p)$;
5. для каждого такого p участник с минимальным номером, стоящий на позиции p, забирает кусочек пиццы толщиной a_p и уходит; a_p увеличивается на 1;
6. после того, как некоторые кусочки пицц забрали, могло случиться так, что у каких-то участников изменилась минимальная толщина кусочка, который они все еще могут получить, в таком случае процесс повторяется с шага 4;
7. иначе, время «запускается заново» и процесс переходит на следующую секунду.

По информации о подходах участников к пиццам определите для каждого участника, кусочек пиццы какой толщины ему достался и сколько времени он провел в ряду.

Формат входных данных

В первой строке ввода через пробел даны целые числа n, m и v — количество пицц, количество участников и время перемещения каждого участника между соседними пиццами ($1 \leqslant n, m \leqslant 10^5$; $1 \leqslant v \leqslant 10^9$).

Во второй строке через пробел перечислены n целых чисел a_i — толщина первого кусочка каждой пиццы ($0 \leqslant a_i \leqslant 10^9$).

В i-й из следующих m строк даны параметры i-го участника: два целых числа s_i и t_i, записанные через пробел — стартовое положение и время подхода участника к ряду пицц ($1 \leqslant s_i \leqslant n$; $0 \leqslant t_i \leqslant 10^9$).

Формат выходных данных

Выведите m строк, в i-й из которых выведите через пробел пару чисел: толщину теста в кусочке пиццы, полученном i-м участником, и время, которое участник потратит на то, чтобы до него дойти.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	**Баллы**	**Ограничения**	**Необходимые подзадачи**
0	–	примеры из условия	
1	8	$\max(t_i) \leqslant 10^4$; $n, m, v \leqslant 100$	0
2	10	все s_i равны и все t_i равны	
3	13	$\lvert a_i - a_j \rvert \geqslant m$ для любых $i \neq j$	
4	19	$n, m \leqslant 1000$	0, 1
5	16	$m \leqslant 1000$	0, 1, 4
6	34	нет	1 – 5

Примеры

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

Замечание

В первом примере события будут происходить в следующем порядке:

1. Первый участник появляется на позиции 5 в момент времени 1; его цель — пицца номер 1.
2. К моменту времени 2 первый участник находится на позиции 4, а на позиции 3 появляется второй участник. Оба будут двигаться к пицце номер 1.
3. В третью секунду не происходит никаких изменений, участники продолжают двигаться влево.
4. В момент времени 4: второй участник стоит около первой пиццы, первый и третий — вместе около второй. Второй учатник забирает кусочек пиццы толщины 1, a_1 становится равно 2.
5. Первый и третий участник моментально видят, что кусочек пиццы толщины 1 они уже не получат, а вот толщины 2 они могут забрать на своей текущей позиции. Первый, как участник с меньшим номером, забирает $a_2 = 2$, после чего a_2 становится равно 3.
6. Оставшемуся третьему участнику остается только сдвинуться еще на 1 влево и забрать $a_1 = 2$.

Задача E. Tree Trisection

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

На территории Университета Иннополис растет множество деревьев, за которыми надо ухаживать. В частности, в Иннополисе растут взвешенные полные двоичные деревья, которые иногда надо подрезать, чтобы они не разрастались очень быстро.

Рассмотрим некоторое полное двоичное дерево с $n = 2^k$ листьями. Дадим корню номер 1, его детям — номера 2 и 3 (слева направо), и так далее занумеруем каждый слой вершин последовательными целыми номерами. Листья дерева будут иметь номера от n до $2n - 1$. У вершины номер i вес равен w_i. Весом множества вершин мы будем называть сумму весов вершин этого множества.

Тогда от вас требуется определять, как эффективно подрезать деревья. Запрос на *подрезку* задается номерами листьев l и r, и осуществляется следующим образом:

1. Сначала выбирается множество вершин V — минимальное по размеру связное множество вершин, содержащее **все** листья с l-го по r-й. Обозначим самую близкую к корню дерева вершину множества V за $\texttt{root}(V)$.

2. Затем требуется выбрать в множестве V две различные вершины x и y такие, что ни одна из них не является потомком другой.

3. Когда вершины x и y выбраны, ребра, ведущие из $\texttt{root}(V)$, x и y «наверх» (в сторону корня), обрезаются. Тем самым V отделяется от всего дерева, и распадается на три подмножества, обозначим их как $V_{\texttt{root}}$, V_x и V_y, соответственно.

4. После чего все три этих подмножества высаживаются в качестве отдельных поддеревьев, чтобы повысить благоустройство территории.

У руководства есть несколько вариантов, как подрезать данное вам дерево. Ваша цель — определить для каждого из их запросов (l, r), какие вершины x и y следует выбрать, чтобы минимизировать максимальный из весов $V_{\texttt{root}}$, V_x и V_y.

Формат входных данных

В первой строке через пробел даны два целых числа n и m — количество листьев в дереве и количество запросов у руководства ($2 \leqslant n \leqslant 2^{16}$; $1 \leqslant m \leqslant 10^5$; $n = 2^k$ для некоторого целого k).

В следующей строке через пробел перечислены $2n - 1$ целых чисел w_i — веса вершин дерева ($1 \leqslant w_i \leqslant 10^9$).

В i-й из следующих m строк дан i-й запрос руководства, два целых числа l_i и r_i — номер первого и последнего листьев, участвующих в подрезке дерева ($n \leqslant l_i, r_i \leqslant 2n - 1$; $r_i \geqslant l_i + 1$).

Формат выходных данных

Для каждого запроса руководства выведите в отдельной строке через пробел два целых числа x и y — номера дополнительных вершин, по которым стоит обрезать дерево, чтобы минимизировать максимальный из весов трех отрезанных частей.

Если есть несколько вариантов ответов, минимизирующих максимальный вес отрезанной части, выведите любой из них.

Система оценки

По умолчанию баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач успешно пройдены. Баллы за подзадачу с потестовой оценкой начисляются за каждый пройденный тест независимо (по 2 балла за каждый из шестнадцати тестов).

Подзадача	**Баллы**	**Ограничения**	**Необходимые подзадачи**	**Оценка**
0	–	примеры из условия		–
1	6	$n \leqslant 32$, $m \leqslant 20$	0	полная группа
2	10	$n \leqslant 512$, $m \leqslant 400$	0, 1	полная группа
3	13	$r_i = l_i + 1$ для всех i		полная группа
4	23	$w_i = 1$ для всех i		полная группа
5	16	$n \leqslant 1024$, $m \leqslant 2000$	0 – 2	полная группа
6	16×2	нет	0 – 5	потестовая

Примеры

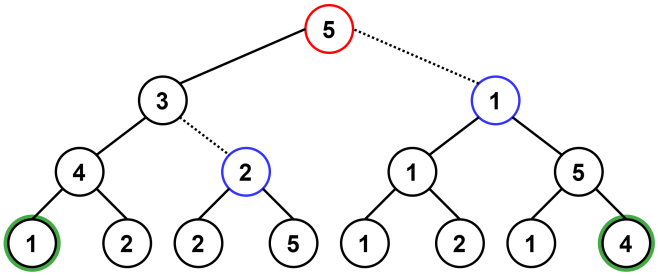
стандартный ввод	стандартный вывод
4 4 1 1 1 1 1 1 1 4 5 5 6 6 7 4 7	4 5 5 3 6 7 5 3
8 7 5 3 1 4 2 1 5 1 2 2 5 1 2 1 4 8 15 8 11 10 13 8 12 11 12 9 14 13 15	5 3 4 11 5 3 4 5 5 3 5 3 6 15

Замечание

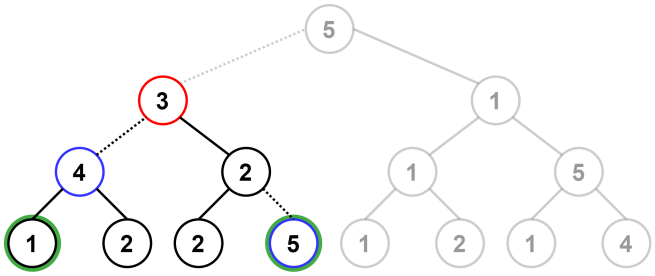
Ниже дана иллюстрация для первых четырех запросов во втором примере из условия задачи.

У каждой вершины отмечен ее вес. Серыми выделены вершины, которые не вошли во множество V. Выбранные листья l и r обведены зеленым. Вершина $\texttt{root}(V)$ обведена красным цветом, а вершины x и y, которые нужно выбрать в качестве ответа — синим. Пунктиром выделены те ребра, которые обрезаются при выполнении соответствующей *подрезки*.

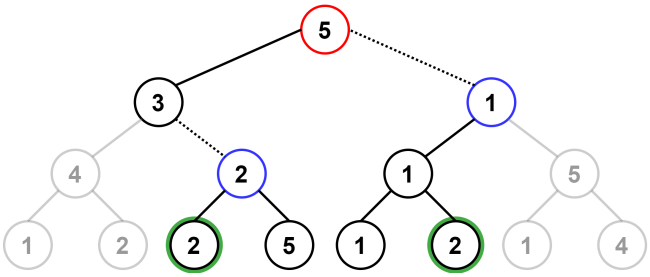
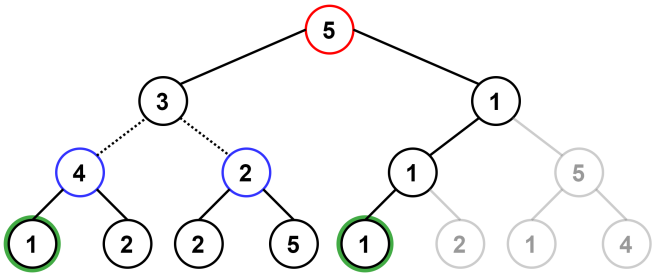
Запрос 1



Запрос 2



Запрос 3



Запрос 4