

Задача А. Лягушка-путешественница

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	512 мегабайт

Лягушонок Горф отправился в путешествие по Болотному королевству. К несчастью, он не рассчитал длину своего прыжка и упал в колодец глубиной n метров. Теперь Горф лежит на самом дне колодца и ему предстоит долгий путь вверх.

Стенки колодца в некоторых местах скользкие, а в некоторых, наоборот, очень удобные. А именно, если Горф сейчас находится на глубине x метров от уровня земли, то за один прыжок он может подняться на любое расстояние от 0 до a_x метров вверх.

К сожалению, Горф не может прыгать без перерывов. После каждого прыжка ему нужно отдохнуть. Если он отдохнет на глубине x метров от уровня земли, то за это время проскользит вниз на b_x метров.

Определите, за какое минимальное число прыжков Горф способен подняться до уровня земли.

Формат входных данных

В первой строке задано целое положительное число n ($1 \leq n \leq 300\,000$) — глубина колодца.

Во второй строке задано n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq i$) — максимальная высота, доступная для прыжка с каждой глубины.

В третьей строке вводится n целых чисел b_1, b_2, \dots, b_n ($0 \leq b_i \leq n - i$) — глубина, на которую лягушонок провалится, взяв перерыв на каждой из глубин.

Формат выходных данных

В первой строке выведите целое число k — минимально возможное количество прыжков. В случае, если лягушонок не может выбраться из колодца, выведите -1 .

Если лягушонок мог выбраться из колодца, то во второй строке выведите последовательность глубин длины k , на которые он будет запрыгивать каждым своим прыжком. Считайте глубину вне колодца в любой точке равной нулю.

Если существует несколько решений, выведите любое из них.

Примеры

стандартный ввод	стандартный вывод
3 0 2 2 1 1 0	2 1 0
2 1 1 1 0	-1
10 0 1 2 3 5 5 6 7 8 5 9 8 7 1 5 4 3 2 0 0	3 9 4 0

Замечание

В первом тесте из условия Горф находится на дне колодца, за один прыжок вверх поднимается к отметке в 1 метр глубины. После этого он соскальзывает вниз на метр и оказывается на отметке в 2 метра. С этой отметки он уже сможет выпрыгнуть из колодца за один прыжок.

Во втором тесте из условия Горф может допрыгнуть до отметки в один метр, но сразу после этого соскользнет обратно на дно колодца, поэтому ему не выбраться.

В третьем тесте из условия выбраться из колодца можно только прыгнув с глубины 5. Попасть напрямую туда нельзя, но можно добраться с помощью серии прыжков $10 \rightarrow 9 \rightarrow 4 \dashrightarrow 5$, где прыжок вверх обозначается простой стрелочкой, а пунктиром обозначен спуск

Задача В. Уничтожение массива

Имя входного файла: стандартный ввод или `input.txt`
Имя выходного файла: стандартный вывод или `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Дан массив a_1, a_2, \dots, a_n , состоящий из целых чисел.

Определим операцию «уничтожения» с целочисленным параметром k ($1 \leq k \leq n$):

- Выбрать k различных индексов массива $1 \leq i_1 < i_2 < \dots < i_k$.
- Вычислить $x = a_{i_1} \text{ AND } a_{i_2} \text{ AND } \dots \text{ AND } a_{i_k}$, где AND это операция побитового «И» (обратитесь к тексту в конце условия для формального определения).
- Вычесть x из всех $a_{i_1}, a_{i_2}, \dots, a_{i_k}$ и оставить остальные элементы массива без изменения.

Найдите все значения k , для которых с помощью конечного количества операций уничтожения с параметром k можно сделать все элементы массива равными 0. Можно показать, что для любого массива a существует хотя бы одно подходящее k .

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следуют описания наборов входных данных.

В первой строке описания каждого набора входных данных находится целое число n ($1 \leq n \leq 200\,000$).

В следующей строке находится n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i < 2^{30}$).

Гарантируется, что сумма n по всем наборам входных данных не превосходит 200 000.

Формат выходных данных

Для каждого набора входных данных выведите все значения k , для которых с помощью операций уничтожения можно сделать все элементы массива равными 0. **Вы должны вывести их в возрастающем порядке.**

Пример

стандартный ввод	стандартный вывод
5	1 2 4
4	1 2
4 4 4 4	1
4	1
13 7 25 19	1 2 3 4 5
6	
3 5 3 1 7 1	
1	
1	
5	
0 0 0 0 0	

Замечание

В первом наборе входных данных:

- При $k = 1$ мы можем сделать четыре операции уничтожения с наборами индексов $\{1\}$, $\{2\}$, $\{3\}$, $\{4\}$. При каждой из этих операций число на соответствующем индексе занулится.
- При $k = 2$ мы можем сделать две операции уничтожения с наборами индексов $\{1, 2\}$, $\{3, 4\}$. При каждой из этих операций числа на соответствующих индексах будут зануляться.

- При $k = 3$ невозможно сделать все a_i равными 0. Если мы сделаем любую операцию уничтожения в самом начале, то в нашем массиве три числа станут равными 0 и одно число останется равным 4. После этого любая операция уничтожения не будет изменять массив, потому что среди индексов, которые используются, будет индекс, значение массива для которого равно 0.
- При $k = 4$ мы можем сделать одну операцию уничтожения с набором индексов $\{1, 2, 3, 4\}$.

Во втором наборе входных данных при $k = 2$ мы можем сделать следующие операции уничтожения:

- Операция с индексами $\{1, 3\}$. Из элементов на этих индексах вычитается $9 = 13 \text{ AND } 25$. Массив после операции станет $[4, 7, 16, 19]$.
- Операция с индексами $\{3, 4\}$. Из элементов на этих индексах вычитается $16 = 16 \text{ AND } 19$. Массив после операции станет $[4, 7, 0, 3]$.
- Операция с индексами $\{2, 4\}$. Из элементов на этих индексах вычитается $3 = 7 \text{ AND } 3$. Массив после операции станет $[4, 4, 0, 0]$.
- Операция с индексами $\{1, 2\}$. Из элементов на этих индексах вычитается $4 = 4 \text{ AND } 4$. Массив после операции станет $[0, 0, 0, 0]$.

Формальное определение побитового «И»

Определим операцию побитового «И» (AND). Пусть даны два целых неотрицательных числа x и y , рассмотрим их двоичные записи (возможно с ведущими нулями): $x_k \dots x_2 x_1 x_0$ и $y_k \dots y_2 y_1 y_0$. Здесь x_i это i -й бит числа x , а y_i это i -й бит числа y . Пусть $r = x \text{ AND } y$ — результат операции AND над числами x и y . Тогда двоичной записью r будет $r_k \dots r_2 r_1 r_0$, где:

$$r_i = \begin{cases} 1, & \text{если } x_i = 1 \text{ и } y_i = 1 \\ 0, & \text{если } x_i = 0 \text{ или } y_i = 0 \end{cases}$$

Задача С. Полёт над озером

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Петя любит делать снимки местности при помощи квадрокоптера. Однажды он отправился в путешествие к живописным берегам озера. Озеро может быть представлено в виде **не обязательно выпуклого** многоугольника на плоскости. Петя запустил квадрокоптер над озером на небольшом расстоянии от поверхности воды и сделал несколько снимков. Если квадрокоптер находится над точкой озера с координатами (x', y') , то на снимке будут видны все точки плоскости, y -координаты которых не превосходят y' . Для каждого снимка Петю интересует суммарная площадь частей озера, которые видны на этом снимке. Но он очень занят обработкой фотографий и просит Вас помочь с этой задачей.

Формат входных данных

В первой строке задано целое число n ($3 \leq n \leq 100\,000$) — количество вершин многоугольника, описывающего озеро.

Каждая из следующих n строк содержит по два целых числа x_i, y_i ($0 \leq x_i, y_i \leq 10^9$) — координаты вершин многоугольника в порядке обхода.

Гарантируется, что озеро представляет собой корректный невырожденный многоугольник, то есть не содержит самопересечений и самокасаний, также никакие три подряд идущие вершины многоугольника не лежат на одной прямой. Обратите внимание, что многоугольник может быть **невыпуклым**.

Следующая строка содержит целое число q ($1 \leq q \leq 100\,000$) — количество снимков.

Каждая из следующих q строк содержит по одному целому числу y'_i ($0 \leq y'_i \leq 10^9$) — y -координату точки озера, над которой находился квадрокоптер, чтобы сделать i -й снимок.

Формат выходных данных

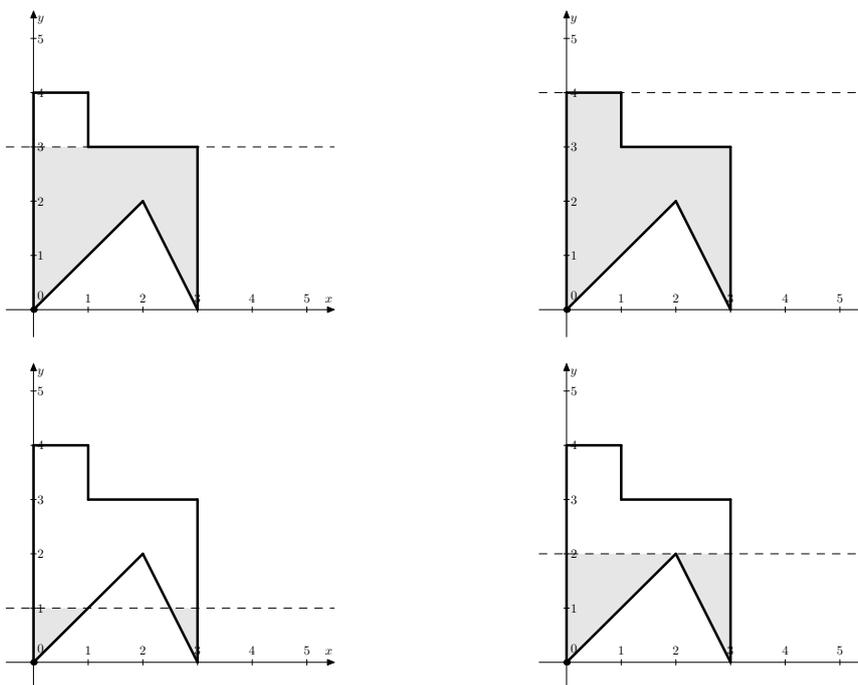
Выведите q чисел s_i — суммарную площадь озера, которую видно на i -й фотографии. Можно доказать, что ответ всегда представим в виде несократимой дроби $\frac{P}{Q}$. Вам нужно вывести $s_i \equiv P \cdot Q^{-1} \pmod{10^9 + 7}$. Иными словами, вам необходимо найти такое s_i ($0 \leq s_i < 10^9 + 7$), что $s_i \cdot Q \equiv P \pmod{10^9 + 7}$.

Примеры

стандартный ввод	стандартный вывод
7 3 0 3 3 1 3 1 4 0 4 0 0 2 2 4 1 3 2 4	750000006 6 3 7
11 5 7 8 7 9 5 12 7 14 7 14 5 12 5 12 2 1 2 1 9 3 9 10 5 8 6 3 13 7 2 1 9 4	33 500000062 45 11 61 55 0 0 61 22

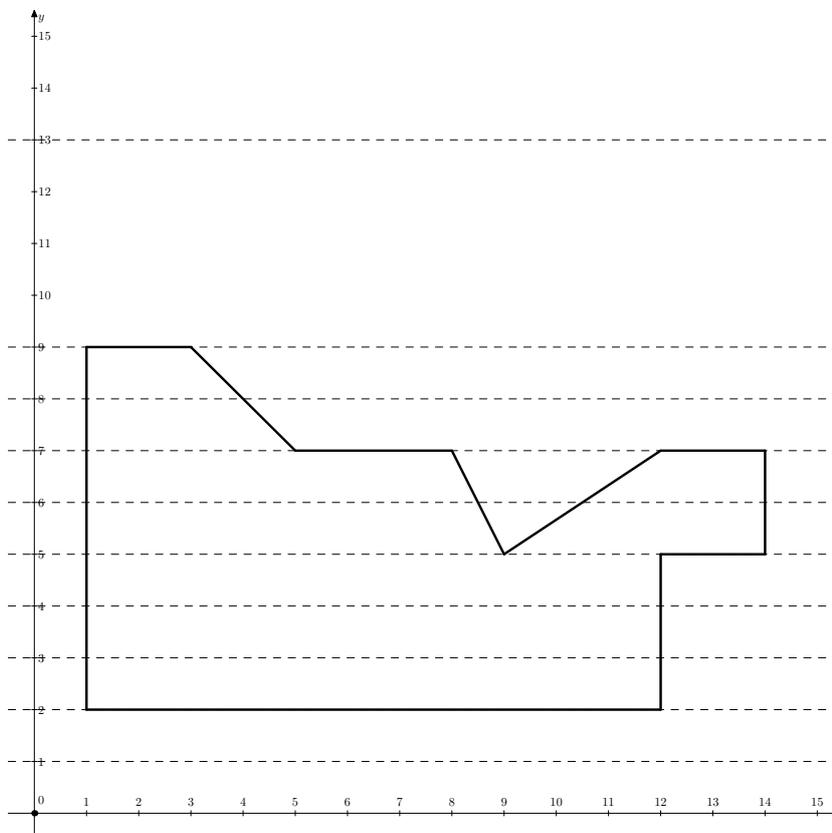
Замечание

В первом тесте из условия озеро и области снимков выглядят так:



На первой фотографии площадь видимой части озера равна $\frac{3}{4}$, на остальных снимках площадь видимой части озера является целым числом.

Во втором тесте из условия озеро и области снимков выглядят так:



На второй фотографии площадь видимой части озера равна $58\frac{1}{2} = \frac{117}{2}$, на остальных снимках площадь видимой части озера является целым числом.

Задача D. Оптимальная вставка

Имя входного файла: стандартный ввод или `input.txt`
Имя выходного файла: стандартный вывод или `output.txt`
Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

У вас есть два массива a_1, a_2, \dots, a_n и b_1, b_2, \dots, b_m , состоящие из целых чисел.

Вы должны вставить все элементы массива b в массив a в произвольные места. В результате получится массив c_1, c_2, \dots, c_{n+m} размера $n + m$.

Обратите внимание, что элементы массива a переставлять нельзя, а элементы массива b можно вставлять куда угодно (в начало, между двумя соседними элементами массива a , в конец) и между собой они могут быть расположены в произвольном порядке.

Какое минимальное количество инверсий в массиве c может быть? Напомним, что инверсией называется пара индексов (i, j) , такая что $i < j$ и $c_i > c_j$.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следуют описания наборов входных данных.

В первой строке описания каждого набора входных данных находятся два целых числа n, m ($1 \leq n, m \leq 10^6$).

Во второй строке описания каждого набора входных данных находятся n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

Во третьей строке описания каждого набора входных данных находятся m целых чисел b_1, b_2, \dots, b_m ($1 \leq b_i \leq 10^9$).

Гарантируется, что сумма n по всем наборам входных данных не превосходит 10^6 и сумма m по всем наборам входных данных не превосходит 10^6 .

Формат выходных данных

Для каждого набора входных данных выведите одно целое число — минимальное количество инверсий, которое может получиться в массиве c при вставке.

Пример

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

Замечание

Варианты оптимальных вставок для всех наборов входных данных (жирным выделены элементы массива a):

- В первом наборе входных данных можно сделать $c = [1, 1, 2, 2, 3, 3, 4]$.
- Во втором наборе входных данных можно сделать $c = [1, 2, 3, 2, 1, 3]$.
- В третьем наборе входных данных можно сделать $c = [1, 1, 3, 3, 5, 3, 1, 4, 6]$.

Задача Е. Онлайн-курс по физкультуре

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Девочка Лола учится в ПТУ, поэтому у нее нет обязательных занятий по физкультуре. Чтобы исправить это недоразумение, Лола решила записаться в спортзал. В спортзале действует система абонементов, в i -й из n дней известна цена покупки абонемента a_i . Разрешается покупать более одного абонемента за день.

Купленный в день i абонемент можно активировать в день i или позднее. Каждый из абонементов после активации будет действовать k дней, то есть активированный в день t абонемент будет действовать в дни с номерами $t, t + 1, \dots, t + k - 1$.

Лола — очень экономная девочка, поэтому она заранее планирует q независимых сценариев похода в спортзал, в i -м сценарии она будет ходить в зал в дни с номерами от l_i до r_i включительно. Лола не станет покупать абонементы до дня l_i . Помогите ей посчитать, какую минимальную сумму придется потратить, чтобы посещать зал в эти дни.

Формат входных данных

В первой строке даны 3 целых числа n, q, k ($1 \leq n, q \leq 300\,000, 1 \leq k \leq n$) — число дней, число сценариев и длительность абонемента.

Во второй строке даны n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

В следующих q строках даны два целых числа l_i, r_i ($1 \leq l_i \leq r_i \leq n$) — описания сценариев.

Формат выходных данных

Для каждого сценария выведите минимальную сумму, необходимую для посещения спортзала.

Пример

стандартный ввод	стандартный вывод
7 5 2	2
2 15 6 3 7 5 6	12
1 2	7
3 7	6
5 5	9
7 7	
3 5	

Замечание

- В первом сценарии Лола купит сертификат в день 1.
- Во втором сценарии она купит один сертификат в день 3 и два сертификата в день 4. Обратите внимание, что она не обязана использовать их в тот же день.
- В третьем сценарии Лола купит сертификат в день 5.
- В четвертом сценарии Лола купит сертификат в день 7.
- В пятом сценарии Лола купит по одному сертификату в дни 3 и 4.

Задача F. Сложная гора

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Группа из n альпинистов подошла к подножию горы. Сложность подъема на эту гору можно оценить целым числом d .

Каждого альпиниста можно описать всего двумя целыми числами s и a , где s характеризует навык альпиниста по восхождению на горы, а a характеризует его аккуратность.

Альпинист с навыком s может забраться на гору сложности p только в том случае, если $p \leq s$. Во время того, как он забирается, он немного меняет путь, по которому идёт, а вместе с ним и его сложность. А именно, после того как на гору сложности p забирается альпинист с аккуратностью a , сложность горы становится равной $\max(p, a)$.

Перед началом восхождения всем стало интересно, какое максимальное количество альпинистов смогут забраться на эту гору, если они будут залезать в оптимальном порядке, а так как в группе только вы увлекаетесь программированием, отвечать на этот вопрос придется вам.

Формат входных данных

В первой строке заданы два целых числа n и d ($1 \leq n \leq 500\,000$, $0 \leq d \leq 10^9$) — количество альпинистов в группе и изначальная сложность горы.

В каждой из последующих n строк содержится по два целых числа s_i и a_i ($0 \leq s_i, a_i \leq 10^9$) — навык восхождению на горы i -го альпиниста и его аккуратность.

Формат выходных данных

Выведите одно число — максимальное количество альпинистов, которые смогут забраться на гору, если они будут подниматься на неё в оптимальном порядке.

Примеры

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

Замечание

В первом примере из условия на гору могут забраться альпинисты 2 и 3 в таком порядке. Других вариантов, при которых забираются два альпиниста, нет.

Во втором примере из условия альпинист 1 забраться не может, потому что изначальная сложность горы слишком велика, а альпинисты 2 и 3 могут забраться в любом порядке.

В третьем примере из условия на гору взберутся альпинисты 5, 3 и 4, причём именно в таком порядке, других вариантов нет.

Задача G. Две сортировки

Имя входного файла: стандартный ввод или `input.txt`
Имя выходного файла: стандартный вывод или `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Целые числа от 1 до n отсортировали по возрастанию в лексикографическом порядке, рассматривая их как строки, и получили массив a_i .

Требуется найти сумму $\sum_{i=1}^n ((i - a_i) \bmod 998244353)$, где $x \bmod y$ означает остаток от деления числа x на число y (этот остаток всегда неотрицателен и не превосходит $y - 1$).

Обратите внимание, что суммирование ведётся без взятия остатка, то есть остаток берётся только от соответствующих разностей. А также обратите внимание, что ответ может не помещаться в 64-битный тип данных.

Формат входных данных

В первой строке содержится одно целое число n ($1 \leq n \leq 10^{12}$).

Формат выходных данных

Выведите одно целое число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
3	0
12	2994733059
21	11978932236
1000000000000	499760084183610382682

Замечание

Число $\overline{x_1x_2\dots x_a}$ меньше числа $\overline{y_1y_2\dots y_b}$ в лексикографическом порядке если верно одно из двух условий:

- либо $a < b$ и $x_1 = y_1, \dots, x_a = y_a$, то есть первое слово является нетривиальным префиксом второго;
- либо есть такая позиция $1 \leq j \leq \min(a, b)$, что $x_1 = y_1, \dots, x_{j-1} = y_{j-1}$ и $x_j < y_j$, то есть, в первой позиции, в которой числа отличаются, в первом числе стоит меньшая цифра.

Например, 42 меньше 6 лексикографически так как числа отличаются в первой позиции и $4 < 6$, $42 < 420$ так как 42 является префиксом 420.

Обозначим 998244353 за M .

В первом примере последовательность a будет равна $[1, 2, 3]$.

$$(1 - 1) \bmod M = 0 \quad \bmod M = 0,$$

$$(2 - 2) \bmod M = 0 \quad \bmod M = 0,$$

$$(3 - 3) \bmod M = 0 \quad \bmod M = 0,$$

$$0 + 0 + 0 = 0.$$

Во втором примере последовательность a будет равна $[1, 10, 11, 12, 2, 3, 4, 5, 6, 7, 8, 9]$.

$$(1 - 1) \bmod M = 0 \quad \bmod M = 0, (2 - 10) \bmod M = (-8) \bmod M = 998244345,$$

$$(3 - 11) \bmod M = (-8) \bmod M = 998244345, (4 - 12) \bmod M = (-8) \bmod M = 998244345,$$

$$(5 - 2) \bmod M = 3 \quad \bmod M = 3, (6 - 3) \bmod M = 3 \quad \bmod M = 3,$$

$$(7 - 4) \bmod M = 3 \quad \bmod M = 3, (8 - 5) \bmod M = 3 \quad \bmod M = 3,$$

$$(9 - 6) \bmod M = 3 \quad \bmod M = 3, (10 - 7) \bmod M = 3 \quad \bmod M = 3,$$

$$(11 - 8) \bmod M = 3 \quad \bmod M = 3, (12 - 9) \bmod M = 3 \quad \bmod M = 3,$$

$$0 + 998244345 + 998244345 + 998244345 + 3 + 3 + 3 + 3 + 3 + 3 + 3 + 3 + 3 = 2994733059.$$

Задача Н. Еще одна игра с фишками

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Петя и Маша никак не могут решить, во что они будут играть. Петя предлагает шахматы, а Маша — в го. Чтобы решить спор, они решили сыграть в следующую игру.

Для этого они взяли поле размером $n \times n$ с двумя фишками на нем. Верхняя левая клетка имеет координаты $(0, 0)$, а правая нижняя — $(n - 1, n - 1)$. Игроки ходят по очереди. Начинает Петя. За один ход нужно переместить **одну** выбранную фишку вверх или влево, иными словами уменьшить **ровно одну** из ее координат на один. При этом перемещать фишку на занятую клетку или вне поля нельзя. Проигрывает тот, кто не может сделать хода.

Сыграв несколько раз они обнаружили, что такая игра не совсем честная, поэтому решили усложнить её и добавили новое правило: нельзя перемещать фишки на клетки, сумма координат которой меньше k .

Ваша задача — узнать, кто выиграет, при оптимальной игре обоих игроков.

Формат входных данных

В первой строке заданы два целых положительных числа n и k ($1 \leq k < n \leq 10^9$) — размеры поля и ограничение на сумму координат клетки с фишкой.

Во второй строке задано одно целое число t ($1 \leq t \leq 100\,000$) — количество игр, для которых нужно узнать, кто выиграет.

Каждая из следующих t строк содержит по четыре целых числа x_1, y_1, x_2, y_2 ($0 \leq x_1, y_1, x_2, y_2 < n; k \leq x_1 + y_1; k \leq x_2 + y_2; x_1 \neq x_2$ или $y_1 \neq y_2$) — исходные координаты фишек.

Формат выходных данных

Для каждой игры, если Петя выигрывает, в новой строке выведите «Petr», иначе выведите «Maria».

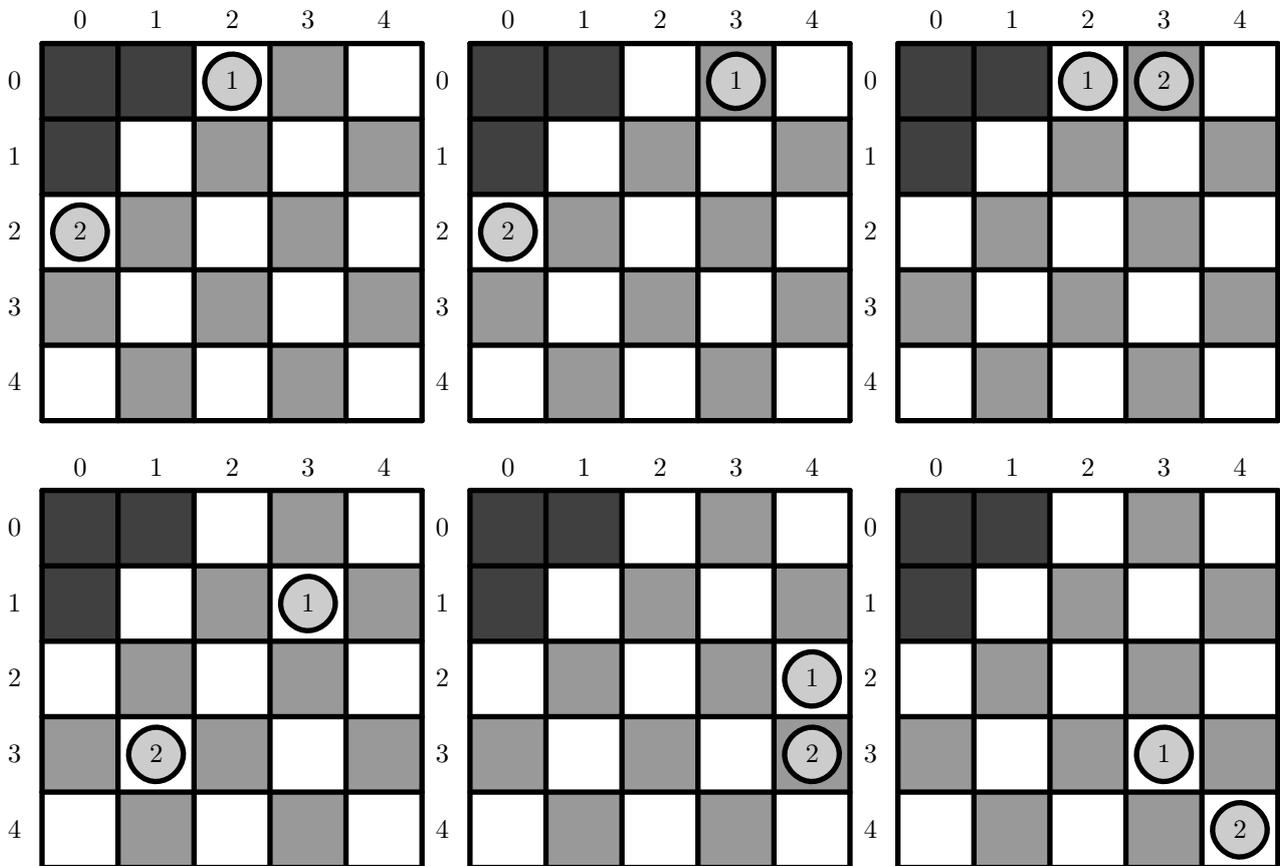
Пример

стандартный ввод	стандартный вывод
5 2	Maria
6	Petr
0 2 2 0	Maria
0 3 2 0	Maria
0 2 0 3	Petr
1 3 3 1	Maria
2 4 3 4	
3 3 4 4	

Замечание

В первой и третьей игре Петя не может сделать ход с самого начала и проигрывает

Во второй игре Петя может сделать единственный ход, после которого Маша ничего не сможет сделать и проиграет.



Задача I. Трудная задача

Имя входного файла: стандартный ввод или `input.txt`
Имя выходного файла: стандартный вывод или `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Дан неориентированный граф, состоящий из $3n$ вершин и **ровно** $3n$ рёбер. Независимым множеством назовём множество вершин такое, что между любыми двумя вершинами в нём нет ребра.

Вам интересно, можно ли найти в данном графе независимое множество, состоящее из n вершин.

Формат входных данных

В первой строке задано целое число n ($1 \leq n \leq 100\,000$).

В следующих $3n$ строках задано описание рёбер графа. Каждое ребро задано парой целых чисел u_i, v_i ($1 \leq u_i, v_i \leq 3n$) — номерами вершин, которые оно соединяет.

Гарантируется, что в графе нет петель и кратных рёбер.

Формат выходных данных

В случае, если в графе нет независимого множества размера n , выведите «No». В противном случае в первой строке выведите «Yes», а во второй — n различных целых чисел — номера вершин в независимом множестве.

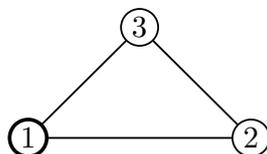
Если существует несколько решений, выведите любое из них.

Примеры

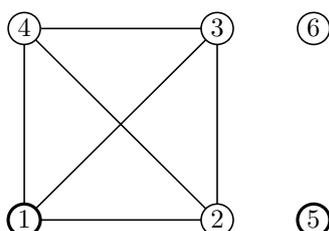
стандартный ввод	стандартный вывод
1 1 2 2 3 3 1	Yes 1
2 1 2 2 3 3 4 4 1 1 3 2 4	Yes 1 5

Замечание

В первом тесте из условия граф выглядит так:



Во втором тесте из условия граф выглядит так:



Задача J. Детский садик «Тормозок»

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

В детском садике «Тормозок» дети решили уравновесить прочные бесконечные качели. Среди них был очень маленький, но очень умный физик Герман, который сразу рассказал всем остальным детям о том, что если они хотят уравновесить качели, то они должны добиться того, чтобы сумма моментов детей, севших слева от центра качелей, была в точности равна сумме моментов детей, севших справа от центра качелей. Для самых юных посетителей детского сада Герман напомнил, что момент ребёнка, севшего на качели, равен произведению его массы на расстояние до центра качелей.

Вам дана масса каждого ребёнка, помогите им выбрать места на качелях так, чтобы они все сидели на целочисленном расстоянии от центра качелей, причём **никакие два ребёнка не сидели на одном месте**, и качели были бы уравновешены. Никакой ребёнок не должен сидеть в центре качелей.

Обозначим за m_i массу i -го ребёнка. Более формально, вам нужно выбрать для каждого ребёнка целое число d_i ($d_i \neq 0$), обозначающее позицию ребёнка на качелях ($d_i > 0$ соответствует расположению ребёнка на расстоянии d_i справа от центра качелей, а $d_i < 0$ — на расстоянии $-d_i$ слева от центра качелей) так, что $\sum_{i=1}^n m_i \cdot d_i = 0$ и все d_i различны.

Формат входных данных

В первой строке задано целое число n ($1 \leq n \leq 1000$) — количество детей.

Во второй строке заданы n целых чисел m_1, m_2, \dots, m_n ($1 \leq m_i \leq 100$), где m_i обозначает массу i -го ребёнка в килограммах.

Формат выходных данных

В случае, если решения нет, выведите «No».

В противном случае, в первой строке выведите «Yes», а во второй строке выведите последовательность n различных целых чисел d_i ($1 \leq |d_i| \leq 10^9$), где d_i обозначает позицию i -го ребёнка. Если $d_i > 0$, то i -й ребёнок должен сесть на расстоянии d_i справа от центра качелей, если же $d_i < 0$, то i -й ребёнок должен сесть на расстоянии $-d_i$ слева от центра качелей.

Если существует несколько решений, выведите любое из них.

Гарантируется, что если решение существует, то существует и решение такое, что $|d_i| \leq 10^9$ для всех i .

Примеры

стандартный ввод	стандартный вывод
1 1	No
2 5 7	Yes 21 -15
5 1 2 3 2 1	Yes 17 -11 28 -5 -69

Замечание

В первом тесте из условия решения не существует.

Во втором тесте из условия сумма моментов равна:

$$5 \cdot 21 + 7 \cdot (-15) = 105 - 105 = 0$$

В третьем тесте из условия сумма моментов равна:

$$1 \cdot 17 + 2 \cdot (-11) + 3 \cdot 28 + 2 \cdot (-5) + 1 \cdot (-69) = 17 - 22 + 84 - 10 - 69 = 0$$