

Задача А. Марио и мировой рекорд

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

Марио — широко известный в узких кругах сантехник. Несмотря на большое разнообразие навыков в сантехнических работах, больше всего Марио нравится передавать воду на большие расстояния. Марио настолько хорошо научился строить длинные водопроводы, что даже решил поставить мировой рекорд по передаче воды на большие расстояния.

Разумеется, в обычных городских условиях построить гигантский водопровод непросто, поэтому Марио взял в аренду у знакомого математика Рене плоский участок бесконечного размера недалеко от города. Чтобы Марио было проще ориентироваться на участке, Рене заботливо ввёл на нём прямоугольную систему координат.

Прибыв на участок, Марио тут же обнаружил небольшой родник в точке с координатами $(0, 0)$ и чрезвычайно обрадовался этому открытию, поскольку родник можно использовать как источник воды для гигантского водопровода.

Для реализации своего проекта Марио купил n стальных труб (не обязательно одинаковых длин), n гибких уголков для труб, позволяющих соединить две трубы под произвольным углом, и кран, который будет расположен в конце водопровода. Пилу, к сожалению, Марио захватить забыл, а ехать через весь бесконечный участок снова ему не хотелось, поэтому сантехник-рекордсмен решил обойтись без распиливания труб на этот раз и использовать трубы только целиком.

Как известно, для надёжности в местах соединения стальных труб гибкими уголками приходится вбивать в землю колышки. Рене не обрадовался желанию Марио вбивать колышки в участок, но как истинный учёный понимал важность эксперимента, поэтому разрешил Марио вбивать колышки для уголков только в точках с целочисленными координатами. В точке установки крана также придётся слегка испортить участок, поэтому кран также можно устанавливать только в точке с целочисленными координатами.

Таким образом, водопровод Марио будет представлять собой ломаную с началом в точке $(0, 0)$ и целочисленными координатами вершин, звеньями которой будут трубы, приобретённые Марио. Обратите внимание, что Марио может не использовать какую-то трубу, но не может использовать трубу дважды. Заметим, что ломаная, в форме которой построит водопровод Марио, может иметь самопересечения и самокасания, однако можно доказать, что оптимальный ответ не содержит ни самопересечений, ни самокасаний.

Марио стремится максимизировать расстояние, на которое он сможет отправить воду из родника, то есть $\sqrt{x_k^2 + y_k^2}$, где (x_k, y_k) — координаты точки, в которой будет установлен кран.

Марио не очень силен в математике, а Рене уже успел уехать по делам, поэтому Марио просит вас помочь ему отправить воду на как можно большее расстояние.

Формат входных данных

В первой строке задано одно целое число n ($1 \leq n \leq 100\,000$) — количество труб, купленных Марио.

Во второй строке заданы n целых чисел a_i ($1 \leq a_i \leq 10^6$), i -е из которых задаёт **квадрат** длины i -й трубы (то есть длина i -й трубы равна $\sqrt{a_i}$).

Формат выходных данных

В первой строке выведите одно целое число k — количество вершин в ломаной, соответствующей оптимальному плану водопровода, включая точки с родником и краном.

В следующих k строках выведите пары целых чисел x_i, y_i — координаты вершин ломаной в порядке от родника до крана.

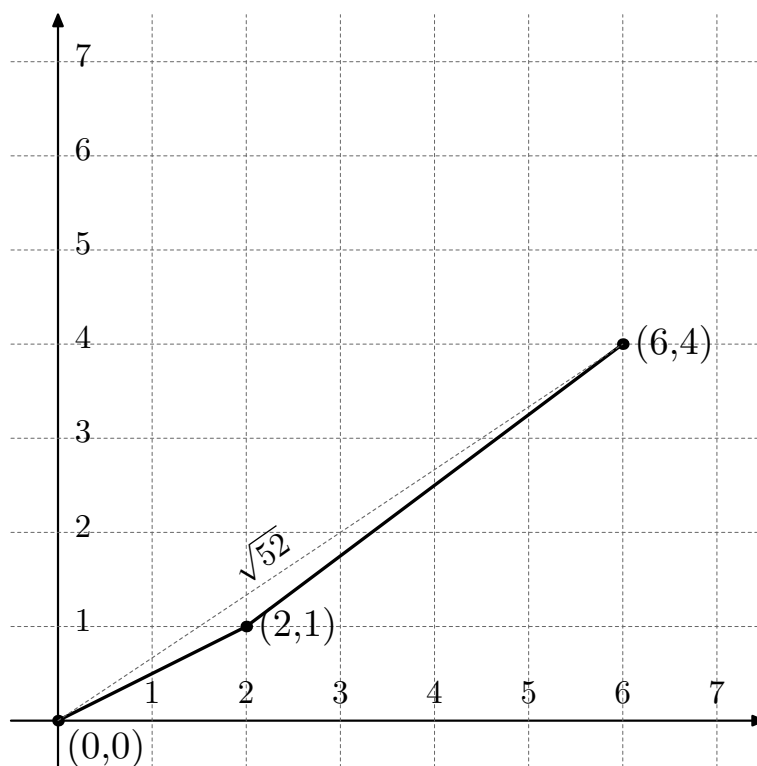
Если ломаных, максимизирующих расстояние от родника до крана, несколько, то выведите любую из них.

Примеры

стандартный ввод	стандартный вывод
2 5 25	3 0 0 2 1 6 4
3 7 3 6	1 0 0
2 1000000 1000000	3 0 0 0 1000 0 2000

Замечание

Пример ответа для первого теста из условия:



В точке $(0,0)$ находится родник. Первая труба длины $\sqrt{5}$ начинается в точке $(0,0)$ с родником и идёт до точки $(2,1)$, в которой при помощи уголка соединяется со второй трубой. Вторая труба длины $\sqrt{25}$ начинается в точке $(2,1)$ и идёт до точки $(6,4)$, в которой находится кран. Таким образом, Марио сможет отправить воду на расстояние $\sqrt{6^2 + 4^2} = \sqrt{52}$.

Задача В. Личность широких взглядов

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

Вася — опытный составитель задач для олимпиад по программированию. На его счету множество разнообразнейших задач, с некоторыми из которых вы можете быть знакомы. Как и все великие творцы, Вася столкнулся с творческим кризисом.

Задачу-то Вася придумал, но не может никак придумать остроумную и уместную легенду к этой задаче. Уже который день Вася не выходит из комнаты, он падает в пучину отчаяния, ужаса, одиночества и безнадежности. Чтобы как-то его развеселить, друг Петя подарил ему строку s , состоящую только из открывающих и закрывающих скобок.

«Ну вот, — подумал Вася, — опять главному герою подарили что-то, чего в реальной жизни никто не дарит. Сейчас, видимо, ещё какое-нибудь свойство припишут этой строке, которое нужно будет вычислить.»

Красотой строки из скобок называется количество её циклических сдвигов, которые являются правильной скобочной последовательностью.

«Вау, как оригинально, — скептически пробурчал себе под нос Вася, — ни разу не видел, чтобы в задачах по программированию требовалось вычислять красоту целого объекта. Ну всё, сейчас мне зачем-то будет надо будет вычислять эту величину на подотрезках и обрабатывать какие-то запросы изменения строки.»

И в самом деле. Для запросов вида (l, r) требовалось определить красоту подотрезка строки s с l по r (включительно). Помимо этого, поступали запросы изменения определённых символов строки s .

«Замечательно, — разочарованно вздохнул Вася, — очередная скучная задача на структуры данных, не хочу её решать. О, а ведь можно дать эту задачу на какую-нибудь олимпиаду!»

Так Вася и поступил. Докажите Васе, что задача совсем не скучная и решите её.

Формат входных данных

В первой строке задано целое число n ($1 \leq n \leq 300\,000$) — длина строки s , которую подарили Васе.

Во второй строке задана строка s длины n , состоящая только из символов «(» и «)».

В третьей строке задано целое число q ($1 \leq q \leq 300\,000$) — количество запросов определения красоты подстроки.

В следующих q строках заданы описания запросов. Запросы бывают двух видов:

- 1 x ($1 \leq x \leq n$). Требуется изменить символ строки s на позиции x на противоположный (то есть (на), а) на ().
- 2 l r ($1 \leq l \leq r \leq n$). Требуется определить красоту подотрезка строки s с символа l по символ r (включительно).

Формат выходных данных

Для запросов второго типа выведите ответы на них в том порядке, в котором они заданы во входном файле.

Пример

стандартный ввод	стандартный вывод
10	2
)((()()())(2
6	0
2 1 10	1
2 3 6	
1 4	
2 2 7	
1 5	
2 3 6	

Замечание

Напомним, что правильной скобочной последовательностью называется скобочная последовательность, которую можно преобразовать в корректное арифметическое выражение путем вставок между ее символами символов «1» и «+». Например, скобочные последовательности «()()», «(())» — правильные (полученные выражения: «(1)+(1)», «((1+1)+1)»), а «)()» и «(» — нет.

Циклическим сдвигом строки s длины n на k ($0 \leq k < n$) называется строка, являющая собой конкатенацию (сложение) последних k символов строки (суффикса длины k) s и первых $n - k$ символов строки (префикса длины $n - k$). Два циклических сдвига на i и j называются различными, если $i \neq j$.

В первом запросе рассматривается строка)((()()())(. У неё есть два циклических сдвига, которые являются правильными скобочными последовательностями: сдвиг на 1 и сдвиг на 9.

Во втором запросе рассматривается строка ()(). У неё также есть два циклических сдвига, которые являются правильными скобочными последовательностями: сдвиг на 0 и сдвиг на 2.

После третьего запроса строка s становится равной)(((()()())(.

В четвёртом запросе рассматривается строка (((()(. Ни один из её циклических сдвигов не является правильной скобочной последовательностью.

После пятого запроса строка s становится равной)((()()())(.

В шестом запросе рассматривается строка ((()). У неё есть только один циклический сдвиг на 0.

Задача С. Иванушка-дурачок и теория вероятностей

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

Недавно Иванушка-дурачок захотел стать умнее и поэтому начал изучать теорию вероятностей. Он считает, что уже достаточно хорошо понимает данную тему, и стал этим хвастаться перед всеми.

Чтобы не быть голословным, Иванушка пообещал продемонстрировать окружающим свои навыки при помощи случайного рисунка. Рисунок представляет собой клетчатое поле с n строками и m столбцами, каждая клетка которого покрашена в белый или чёрный цвет. Иванушка считает, что рисунок случайный, если для любой клетки верно, что среди ее соседей по стороне не более одной клетки такого же цвета, что и она.

Его братья быстро выяснили, что задумал Иванушка. Они долго пытались объяснить ему, что он, дурачок, все ещё плохо понимает, что такое случайность. Но Иванушка так и не поверил им, поэтому они попросили помощи у вас. Они хотят найти количество различных случайных по мнению Иванушки рисунков. Два рисунка называются различными если существует клетка, которая покрашена в разные цвета на этих рисунках. Так как это число может быть слишком большим, найдите остаток от деления его на $10^9 + 7$.

Формат входных данных

В единственной строке заданы два целых числа n и m ($1 \leq n, m \leq 100\,000$) — количество строк и столбцов клетчатого поля, соответственно.

Формат выходных данных

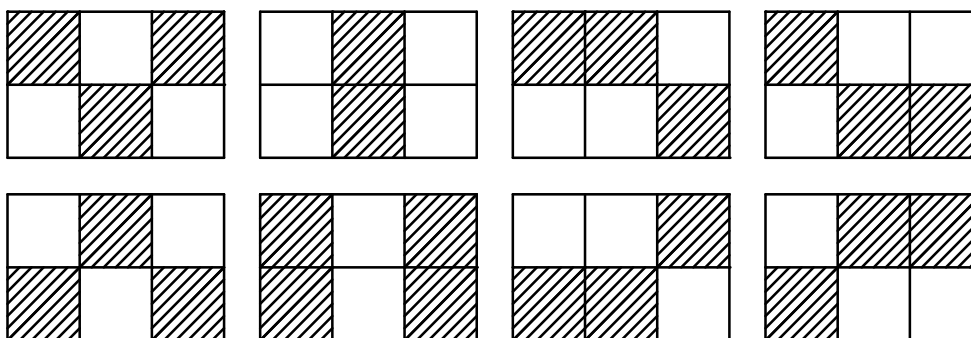
Выведите одно целое число — остаток от деления числа случайных рисунков на $10^9 + 7$.

Пример

стандартный ввод	стандартный вывод
2 3	8

Замечание

На картинке ниже изображены все возможные случайные раскраски поля 2 на 3.



Задача D. Дороги в стране

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

В некоторой стране есть n городов, пронумерованных целыми числами от 1 до n . Столица этой страны расположена в городе с номером 1. Некоторые пары городов в этой стране соединены дорогами с односторонним движением, при этом для каждой пары городов u и v есть не более одной дороги, ведущей из города u в город v , однако наличие дороги из города u в город v не запрещает существование дороги из города v в город u . Для каждой дороги известно, груз какого максимального веса можно по ней перевозить. Требуется для каждого города определить максимальный вес груза, который можно доставить из него в столицу.

Формат входных данных

В первой строке содержатся два целых числа n и m ($2 \leq n \leq 10^5$, $0 \leq m \leq 10^6$) — количество городов и дорог в стране, соответственно.

В следующих m строках содержатся описания дорог. Описание дороги состоит из трёх целых чисел u , v и w ($1 \leq u, v \leq n$, $u \neq v$, $1 \leq w \leq 10^9$), описывающих дорогу с односторонним движением из города с номером u в город с номером v , по которой можно перевозить грузы весом не более w . Гарантируется, что для каждой пары городов u и v существует не более одной дороги из u в v .

Формат выходных данных

Выведите $n - 1$ целых чисел. i -е число должно быть равно максимальному весу груза, который можно перевезти из города $i + 1$ в столицу (город с номером 1). Если из города $i + 1$ в столицу доставить груз невозможно, то выведите -1 .

Примеры

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

Замечание

В первом примере есть только одна дорога, заканчивающаяся в столице, и по ней нельзя везти грузы весом более 1, при этом из всех городов можно доехать до столицы по дорогам. Таким образом, из всех городов можно привезти в столицу груз весом не более 1.

Во втором примере из города 2 в город 1 можно перевезти груз весом 3 следующим образом: проехать из города 2 в город 3 по дороге, по которой можно везти грузы весом не более 3, а затем из города 3 в столицу (город 1) по дороге, по которой можно везти грузы весом не более 7. Из города 3 в город 1 можно доставить груз весом 7 по прямой дороге, соединяющей эти города. Аналогично, из города 4 в город 1 можно перевезти груз весом 1 по дороге, соединяющей эти города напрямую.

В третьем примере из города 3 в столицу доехать нельзя.

Задача Е. Очередь в поезде

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 вагоне поезда на n местах сидят n людей, человек с номером i сидит на месте с номером i . Места пронумерованы слева направо от 1 до n . Поездка долгая, поэтому каждый человек в какой-то момент проголодается и захочет заварить лапшу быстрого приготовления кипятком, i -й человек захочет выйти за кипятком ровно в начале t_i -й минуты.

Бак с кипятком находится левее 1-го места за дверью. Поскольку баком одновременно может пользоваться не более одного человека, люди стоящие у бака образуют очередь в порядке, в котором они подошли к баку. Как только человек заканчивает наливать кипяток, следующий за ним в очереди человек (если такой есть) начинает наливать кипяток себе. Каждый человек наливает себе кипяток ровно p минут.

Люди не любят просто так стоять в очереди. После того, как человек на месте с номером i захочет сходить за кипятком, в начале каждой минуты (включая t_i -ю минуту) он будет оценивать размер очереди, в которую он попадёт, если отправится за кипятком прямо сейчас, а именно он посмотрит на все места с номерами от 1 до $i - 1$. Если хотя бы одно из этих мест пусто, то есть в очереди возле бака есть люди с номерами, меньшими i , человек с номером i решит, что очередь слишком большая и останется на своём месте до начала следующей минуты. В противном случае человек с номером i тут же встанет со своего места и приготовится идти за кипятком. Однако если человек с номером j ($j < i$) также решит отправиться за кипятком в этот же момент, человек с номером i это увидит, сядет обратно на своё место и снова будет ждать начала следующей минуты. Таким образом, в начале каждой минуты не более одного человека отправится за кипятком. Обратите внимание, что конструкция вагона такова, что люди видят только места перед собой, но не видят бак и очередь перед ним, поэтому могут принимать решения только исходя из наличия людей на местах перед ними.

Временем перемещения людей по вагону можно пренебречь. Например, если человек проголодался в начале t -й минуты, и в этот момент очередь к баку была пуста, то он окажется у бака в начале t -й минуты, закончит наливать воду в конце минуты с номером $t + p - 1$, и в начале минуты с номером $t + p$ уже будет сидеть на своём месте.

Для каждого человека определите, на какой минуте он вернётся на своё место с кипятком для лапши.

Формат входных данных

Первая строка содержит два целых числа n и p ($1 \leq n \leq 100\,000$, $1 \leq p \leq 10^9$) — количество людей в вагоне и количество минут, которое каждый человек тратит на использование бака.

Во второй строке содержатся n целых чисел t_1, t_2, \dots, t_n ($0 \leq t_i \leq 10^9$), i -е из которых означает, что i -й человек проголодается в начале t_i -й минуты.

Формат выходных данных

Выведите n целых чисел, i -е из которых — номер минуты, на которой i -й человек вернётся на своё место с кипятком для лапши.

Пример

стандартный ввод	стандартный вывод
5 314	314 628 1256 942 1570
0 310 942 628 0	

Замечание

В примере в 0-ю минуту за кипятком одновременно захотели пойти 1-й и 5-й человек, первый вышел сразу и на 314-й минуте вернулся от бака. За это время множество желающих выйти за кипятком пополнилось 2-м человеком, и следующим пошел 2-й человек и т.д. 5-й человек в итоге вышел последним.

Задача F. XOR шифрование

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

Алиса и Боб в очередной раз решают важные криптографические вопросы. На этот раз они хотят научиться передавать информацию, вызывая как можно меньше подозрений о её зашифрованности.

Канал, по которому Алиса и Боб обмениваются информацией, прослушивает Ева. Алиса может прислать Бобу некоторое множество A целых неотрицательных чисел. Если для некоторого достаточно большого $t \geq 0$ все числа $0, 1, 2, \dots, t$ лежат в множестве A , то это повышает подозрения Евы о том, что переданная информация была зашифрована перед отправкой. Поэтому передавая некоторое множество A целых чисел, Алиса хочет минимизировать минимальное целое неотрицательное число, которое не принадлежит множеству A , назовем это число *тревожностью* Евы.

Для того, чтобы зашифровать информацию, Алиса с Бобом придумали такую простую схему: пусть Алиса хочет передать Бобу множество $A = \{a_1, a_2, \dots, a_n\}$ целых неотрицательных чисел. Алиса может выбрать некоторое целое число x ($0 \leq x \leq k$) и передать Бобу множество $\{a_1 \oplus x, a_2 \oplus x, \dots, a_n \oplus x\}$, где значок \oplus обозначает операцию взятия побитового исключающего ИЛИ двух чисел. При этом о значении числа k они договорились заранее. Таким образом, Алиса хочет сделать свой выбор так, чтобы тревожность Евы, а именно минимальное целое неотрицательное число, не принадлежащее множеству $\{a_1 \oplus x, a_2 \oplus x, \dots, a_n \oplus x\}$, было как можно меньше.

Алиса еще не выбрала, какое конкретно множество A она хочет передать Бобу. Изначально она хочет передать Бобу пустое множество $A = \emptyset$. После этого Алиса q раз принимает решение о том, что хочет добавить или удалить из текущего множества A некоторое число. И после каждой такой операции она хотела бы знать минимальную возможную тревожность Евы, которой Алиса может добиться, если она захочет передать Бобу текущее множество. Помогите ей и напишите программу, которая будет вычислять это значение.

Формат входных данных

В первой строке находятся два целых числа q и k ($1 \leq q \leq 200\,000, 0 \leq k < 2^{20}$) — количество изменений множества Алисы и верхнее ограничение на x , которое может выбирать Алиса.

В следующих q строках находится по два целых числа, разделенных пробелом, в i -й строке находятся $type_i, x_i$ ($type_i \in \{1, 2\}, 0 \leq x_i < 2^{20}$). Если $type_i = 1$, это означает, что в i -м изменении Алиса хочет добавить x_i в множество A , которое она сейчас собирает отправлять Бобу, если $type_i = 2$, это означает, что в i -м изменении Алиса хочет удалить x_i из множества A , которое она сейчас собирает отправлять Бобу. Гарантируется, что если $type_i = 1$, то числа x_i нет в текущем множестве A , если $type_i = 2$, то число x_i есть в текущем множестве A .

Формат выходных данных

Выведите q строк, в i -й строке ($1 \leq i \leq q$) выведите минимально возможное значение тревожности Евы, которого может добиться Алиса, когда будет пересылать множество A , которое получается у нее после первых i изменений.

Примеры

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

Замечание

Рассмотрим первый пример.

После первого запроса $A = \{1\}$. Тогда, выбрав $x = 0$, можно получить множество $\{1 \oplus 0\} = \{1\}$. Минимальное неотрицательное целое число, не лежащее в этом множестве, равно 0.

После второго запроса $A = \{0, 1\}$. Тогда, выбрав $x = 2$, можно получить множество $\{0 \oplus 2, 1 \oplus 2\} = \{2, 3\}$. Минимальное неотрицательное целое число, не лежащее в этом множестве, равно 0.

После третьего запроса $A = \{0, 1, 2\}$. Тогда, выбрав $x = 2$, можно получить множество $\{0 \oplus 2, 1 \oplus 2, 2 \oplus 2\} = \{0, 2, 3\}$. Минимальное неотрицательное целое число, не лежащее в этом множестве, равно 1.

После четвёртого запроса $A = \{0, 2\}$. Тогда, выбрав $x = 1$, можно получить множество $\{0 \oplus 1, 2 \oplus 1\} = \{1, 3\}$. Минимальное неотрицательное целое число, не лежащее в этом множестве, равно 0.

После пятого запроса $A = \{0, 2, 3\}$. Тогда, выбрав $x = 1$, можно получить множество $\{0 \oplus 1, 2 \oplus 1, 3 \oplus 1\} = \{1, 2, 3\}$. Минимальное неотрицательное целое число, не лежащее в этом множестве, равно 0.

После шестого запроса $A = \{0, 1, 2, 3\}$. Тогда, выбрав $x = 0$, можно получить множество $\{0 \oplus 0, 1 \oplus 0, 2 \oplus 0, 3 \oplus 0\} = \{0, 1, 2, 3\}$. Минимальное неотрицательное целое число, не лежащее в этом множестве, равно 4.

После седьмого запроса $A = \{0, 1, 2, 3, 4\}$. Тогда, выбрав $x = 1$, можно получить множество $\{0 \oplus 1, 1 \oplus 1, 2 \oplus 1, 3 \oplus 1, 4 \oplus 1\} = \{0, 1, 2, 3, 5\}$. Минимальное неотрицательное целое число, не лежащее в этом множестве, равно 4.

После восьмого запроса $A = \{0, 1, 2, 4\}$. Тогда, выбрав $x = 2$, можно получить множество $\{0 \oplus 2, 1 \oplus 2, 2 \oplus 2, 4 \oplus 2\} = \{0, 2, 3, 6\}$. Минимальное неотрицательное целое число, не лежащее в этом множестве, равно 1.

Задача G. Целые точки

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

DLS и JLS скучают на уроке математики в школе. Чтобы как-то развлечь себя, DLS достал лист клетчатой бумаги и нарисовал на ней n различных прямых, заданных уравнениями $y = x + p_i$, для каждого i до 1 до n . JLS не заставил себя ждать и нарисовал на листе DLS m различных прямых, заданных уравнениями $y = -x + q_i$, для каждого i до 1 до m . DLS и JLS интересно, сколько пар прямых пересекаются в точках с целочисленными координатами. К сожалению, до конца урока осталось мало времени, поэтому DLS и JLS просят вас помочь им.

Формат входных данных

В первой строке задано целое число n ($1 \leq n \leq 100\,000$) — количество прямых, которые нарисовал DLS.

Во второй строке содержатся n различных целых чисел p_i ($0 \leq p_i \leq 10^9$), описывающих прямые, нарисованные DLS. Каждое из чисел описывает прямую, заданную уравнением $y = x + p_i$.

В третьей строке задано целое число m ($1 \leq m \leq 100\,000$) — количество прямых, которые нарисовал JLS.

В четвёртой строке содержатся m различных целых чисел q_i ($0 \leq q_i \leq 10^9$), описывающих прямые, нарисованные JLS. Каждое из чисел описывает прямую, заданную уравнением $y = -x + q_i$.

Формат выходных данных

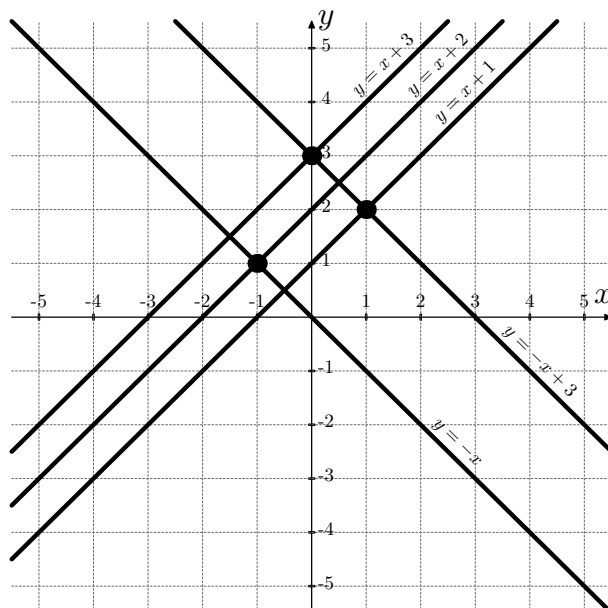
Выведите одно целое число — количество пар прямых, которые пересекаются в точке с целочисленными координатами.

Пример

стандартный ввод	стандартный вывод
3	3
1 3 2	
2	
0 3	

Замечание

Ниже изображены прямые из первого примера. Чёрными кружочками выделены целочисленные точки пересечения.



Задача Н. Фокус с делением и умножением

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

Наверняка вы знаете много математических фокусов. В этой задаче вам предлагается продемонстрировать свои умения в одном из таких фокусов.

У вас есть массив из различных натуральных чисел a_1, a_2, \dots, a_n . Помимо этого, перед вами стоит стол, на котором лежат 10^{18} карточек, на которых записаны все натуральные числа от 1 до 10^{18} , каждое ровно на одной карточке.

Вы можете подойти к столу и взять карточку с числом x . После этого вы можете выбрать ровно одно любое число a_i в вашем массиве и поменять его либо на $\frac{a_i}{x}$, либо на $a_i \cdot x$. При этом делить a_i на x разрешается только в случае, если a_i делится на x . Таким образом все числа в вашем массиве должны оставаться натуральными. После выполнения данной операции использованная карточка выкидывается и больше никогда не возвращается на стол.

Вы хотите несколько раз взять со стола карточки и выполнить операции таким образом, чтобы в итоге все числа массива оказались равными. Определите, какое минимальное количество карточек потребуется взять со стола, чтобы совершить этот фокус. Гарантируется, что хотя бы один способ сделать все числа в массиве равными существует.

Формат входных данных

В первой строке находится одно целое число n ($1 \leq n \leq 200\,000$) — количество чисел в вашем массиве.

Во второй строке находятся n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_1 < a_2 < \dots < a_n \leq 10^{18}$), разделенных пробелами. Для удобства элементы массива даны в возрастающем порядке.

Формат выходных данных

Выведите минимальное количество карточек, которое нужно взять со стола, чтобы сделать все числа массива равными с помощью операций деления и умножения, описанных в условии задачи.

Примеры

стандартный ввод	стандартный вывод
2 1 2	1
5 2 3 6 12 18	5
3 239 717 1434	2

Замечание

В первом тесте, можно подойти к столу и взять карточку, на которой написано число 2. После этого можно выбрать элемент массива $a_2 = 2$ и заменить его на $\frac{2}{2} = 1$. Таким образом, все числа массива станут равны 1.

Во втором тесте можно подойти к столу пять раз и брать карточки, на которых записаны числа 32, 24, 12, 6, 4 и умножать на них первый, второй, третий, четвертый и пятый элемент массива, соответственно. Тогда все числа массива станут равны 72.

В третьем тесте можно подойти к столу, взять карточку с числом 3 и умножить a_1 на него. После этого можно взять карточку с числом 2 и поделить a_3 на него. Таким образом получится, что все числа массива равны 717.

Задача I. Жулик, не воруй

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

Я Карта! Я КАРТа! Я КААААРТА!!!

Карта

В ожидании новых приключений Башмачок захотел совершить доброе дело. Посоветавшись с Картой и Рюкзаком, он решил подарить Даше связный граф. Даша не любит, когда в графе присутствуют петли или кратные ребра, и Башмачок прекрасно об этом знает. После долгих поисков Башмачок выбрал t вариантов графа, которые могли бы понравиться Даше, но, к сожалению, все его планы решил испортить лис Жулик.

Жулик узнал, что Даша недавно научилась считать до 3, и у него родилась идея. Он хочет совершить кражу так, чтобы Даша не заметила пропажи, поэтому он решил украсть непустой набор вершин такой, что если удалить выбранные вершины и связанные с ними рёбра из графа, то для любой из оставшихся вершин будет верно, что ее степень (число смежных рёбер) будет иметь такой же остаток от деления на 3, что и степень до удаления рёбер. Также он понял, что будет слишком подозрительно, если он украдет все вершины, поэтому он решил воздержаться от такой соблазнительной затеи.

Башмачок понял, что он никак не может позволить случиться преступлению. Но он вскоре осознал, что одному ему не справиться. Поэтому он решил попросить у вас помощи. Башмачок хочет для каждого варианта графа узнать, сможет ли Жулик осуществить свои злобные планы.

Формат входных данных

В первой строке задано целое число t ($1 \leq t \leq 100\,000$) — количество вариантов графа.

Первая строка описания каждого варианта содержит два целых числа n , m ($1 \leq n \leq 500\,000$, $0 \leq m \leq 500\,000$) — количество вершин и рёбер в графе, соответственно.

Затем следуют m строк, в каждой из которых записаны два целых числа a_i , b_i ($1 \leq a_i, b_i \leq n$) — номера вершин, соединяемых i -м ребром. Гарантируется, что граф связан и что в нем нет петель и кратных ребер.

Гарантируется, что сумма n по всем вариантам не превосходит 500 000, и что сумма m по всем вариантам не превосходит 500 000.

Описания вариантов графа разделены одной пустой строкой.

Формат выходных данных

Для каждого из вариантов графа выведите «Yes», если Жулик может осуществить свой план для данного варианта графа, или «No», если Жулику не удастся обмануть Дашу.

В случае, если Жулик может осуществить свой коварный план, выведите пример, как именно он может это сделать. Первая строка примера должна содержать целое число c ($1 < c < n$) — количество вершин, которые может украсть Жулик, чтобы Даша не заметила пропажи. В следующей строке выведите c различных чисел — номера этих вершин. Порядок вывода вершин не важен. Если способов украсть вершины требуемым образом несколько, выведите любой из них.

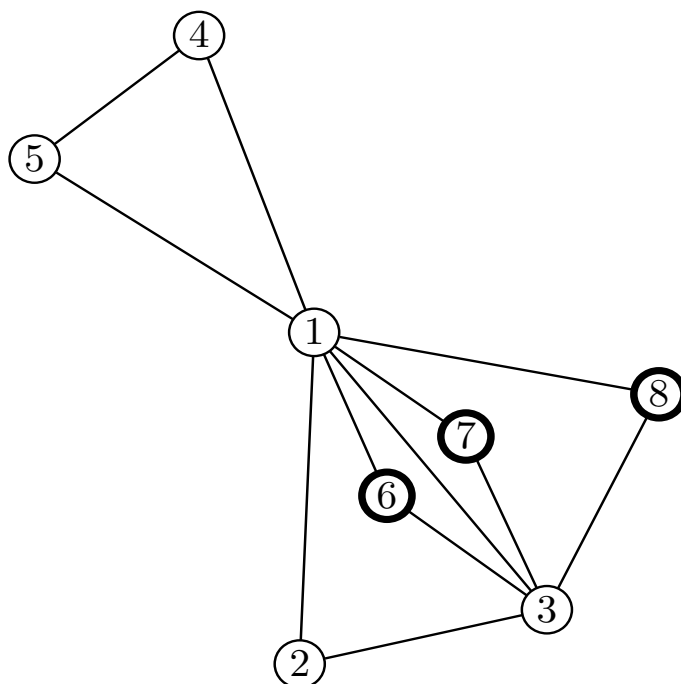
Обратите внимание, что Жулик необязательно крадет максимальное возможное число вершин.

Пример

стандартный ввод	стандартный вывод
3	No
3 3	Yes
1 2	3
2 3	4 5 6
3 1	Yes
	3
6 6	6 7 8
1 2	
1 3	
2 3	
2 5	
2 6	
2 4	
8 12	
1 2	
1 3	
2 3	
1 4	
4 5	
5 1	
3 6	
3 7	
3 8	
6 1	
7 1	
8 1	

Замечание

На картинке ниже изображен третий вариант из примера. Жирным отмечены вершины, которые может украсть Жулик.



Задача J. Конкурс котиков

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

В Котовице в ближайшие выходные будет проходить конкурс котиков. Для конкурса нужно выбрать жюри и участников. Всего в Котовице n жителей, и у каждого жителя живет ровно один котик. Жители и котики пронумерованы целыми числами от 1 до n , причем у i -го жителя живёт i -й котик.

Каждый житель Котовице знаком с несколькими котиками, включая, конечно же, своего. Для конкурса нужно выбрать нескольких жителей на роль жюри, и нескольких котиков на роль участников. Для того чтобы конкурс состоялся, в нём должен принять участие хотя бы один член жюри, и хотя бы один участник. Для того чтобы конкурс прошёл честно, ни один член жюри не должен быть знаком ни с одним участником. И, наконец, чтобы конкурс прошёл наиболее интересно, было решено, что количество членов жюри плюс количество участников должно равняться n .

Помогите жителям Котовице выбрать состав жюри и участников для предстоящего конкурса, либо определите, что это сделать невозможно.

Формат входных данных

В первой строке задано одно целое число t ($1 \leq t \leq 100\,000$) — количество тестовых случаев. Затем следует описание t тестовых случаев, каждый из которых задан следующим образом:

В первой строке заданы два целых числа n и m ($1 \leq n \leq m \leq 10^6$) — количество жителей в Котовице и количество пар знакомых жителей и котиков, соответственно.

В следующих m строках заданы пары целых чисел a_i и b_i ($1 \leq a_i, b_i \leq n$), обозначающих, что a_i -й житель знаком с b_i -м котиком. Каждая пара жителя и котика встречается в списке знакомых не более одного раза. Гарантируется, что для любого i в списке присутствует пара из i -го жителя и i -го котика.

Тестовые случаи разделены одной пустой строкой.

Гарантируется, что сумма n по всем тестам не превосходит 10^6 , и сумма m по всем тестам не превосходит 10^6 .

Формат выходных данных

Для каждого теста в единственной строке выведите «No», если выбрать состав жюри и участников для конкурса невозможно. Иначе, в первой строке выведите «Yes». Во второй строке выведите два целых числа j и p ($1 \leq j, 1 \leq p, j + p = n$) — количество членов жюри и участников в конкурсе, соответственно. В третьей строке выведите j различных целых чисел от 1 до n — номера жителей, которые должны быть выбраны на роль жюри. В четвертой строке выведите p различных целых чисел от 1 до n — номера котиков, которые должны быть выбраны на роль участников. Если различных подходящих ответов несколько, можно вывести любой из них.

Пример

стандартный ввод	стандартный вывод
4	Yes
3 4	2 1
1 1	1 3
2 2	2
3 3	Yes
1 3	1 2
	2
3 7	1 3
1 1	No
1 2	No
1 3	
2 2	
3 1	
3 2	
3 3	
1 1	
1 1	
2 4	
1 1	
1 2	
2 1	
2 2	

Замечание

В первом тесте на роль жюри можно выбрать первого и третьего жителя, ни один из них не знаком со вторым котиком, поэтому его можно выбрать на роль участника.

Во втором тесте на роль жюри можно выбрать второго жителя, он не знаком ни с первым, ни с третьим котиком, которых можно выбрать на роль участников.

В третьем тесте ответа не существует, потому что единственный житель знаком с единственным котиком. Поэтому они не могут участвовать в конкурсе одновременно. Значит, в конкурсе не будет участвовать ни один житель, либо ни один котик.

В четвертом тесте, аналогично третьему, каждый житель знаком с каждым котиком, поэтому в конкурсе не могут одновременно участие какой-то житель и какой-то котик.

Задача К. Черепашка

Имя входного файла: стандартный ввод или `input.txt`
Имя выходного файла: стандартный вывод или `output.txt`
Ограничение по времени: 5 секунд
Ограничение по памяти: 512 мегабайт

У мальчика Коля есть черепашка и поле $2 \times n$. Строки поля пронумерованы от 1 до 2 сверху вниз, а столбцы от 1 до n слева направо.

Предположим, что в каждой клетке есть съедобный лист салата, энергетическая ценность которого в клетке поля в строке i и столбце j составляет $a_{i,j}$. Черепашка изначально находится в верхней левой клетке и хочет попасть в правую нижнюю. Черепашка умеет двигаться только вниз и вправо, а среди всех возможных путей она выберет тот, который максимизирует суммарную энергетическую ценность листьев салата, через которые она проходит (если таких путей несколько, она выберет произвольный из них).

Коля беспокоится, что если черепашка съест слишком много салата, то это может быть опасно для её здоровья. Поэтому он хочет переставить листья салата в таблице таким образом, чтобы минимизировать суммарную энергетическую ценность салата, который съест черепашка. Обратите внимание, что минимизировать количество перестановок листьев салата не требуется, то есть Коля просто соберёт все листья со всех клеток и разложит их заново, так чтобы в каждой клетке оказался ровно один лист.

Формат входных данных

Первая строка содержит одно целое число n ($2 \leq n \leq 25$) — длину поля.

Вторая строка содержит n целых чисел $a_{1,i}$ ($0 \leq a_{1,i} \leq 50\,000$), описывающих энергетическую ценность листьев салата в первой строке поля.

Третья строка содержит n целых чисел $a_{2,i}$ ($0 \leq a_{2,i} \leq 50\,000$), описывающих энергетическую ценность листьев салата во второй строке поля.

Формат выходных данных

Выведите две строки по n чисел в каждой — расстановку листьев салата после переупорядочивания Коли.

Если существует несколько оптимальных переупорядочиваний — выведите любое из них.

Примеры

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

Замечание

В первом примере черепашка после перестановки сможет съесть салат суммарной энергетической ценности $1 + 4 + 2 = 7$.

Во втором примере черепашка съест салат суммарной энергетической ценности 0.

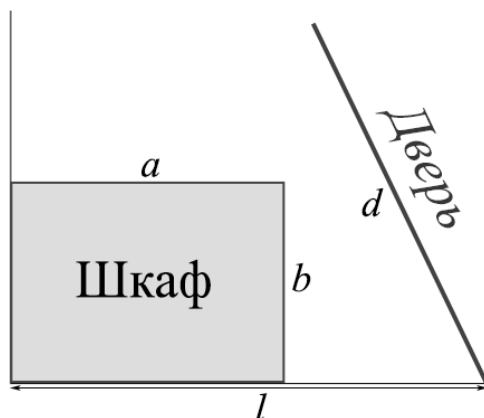
В третьем примере черепашка сможет съесть салат суммарной энергетической ценности 1.

Задача L. Дорогой шкаф

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

Владимир — успешный молодой человек, который всего за 3 года смог накопить достаточно денег и купить себе квартиру. Более того, у него осталась приличная сумма, чтобы купить себе в квартиру дорогой зеркальный шкаф. После долгих хождений по магазинам и поискам приличной мебели, Владимир наконец нашёл подходящий ему шкаф, имеющий форму прямоугольного параллелепипеда с длиной основания a и шириной основания b .

Владимир хочет поставить шкаф в угол своей комнаты прямо за дверью. Но вот незадача, дверь в квартире еще не установлена. Теперь Владимиру интересно, сможет ли дверь при открытии не задеть шкаф и упереться в стену, ведь в противном случае дорогое зеркало на шкафу может поцарапаться. Зная длину двери d , а также расстояние от двери до угла l , помогите Владимиру это выяснить. Даже если дверь немного коснётся шкафа, она его поцарапает, и в этом случае считается, что дверь задела шкаф. Дверь может упереться только в шкаф или в одну из двух стен, стыкующихся в углу шкафа под прямым углом. Две другие стены и вся остальная мебель находятся достаточно далеко. Для лучшего понимания ниже показан вид сверху на угол комнаты.



Формат входных данных

В четырёх строках заданы четыре целых числа a , b , d и l ($1 \leq a, b, d, l \leq 30\,000$, $a \leq l$) — длина шкафа, ширина шкафа, длина двери и расстояние от двери до угла комнаты, соответственно.

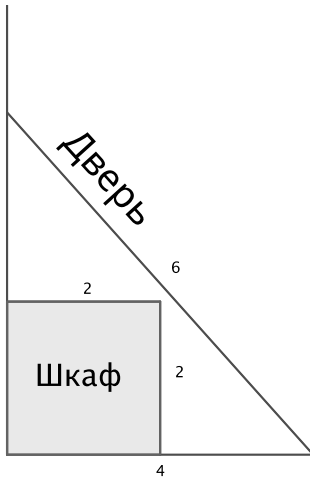
Формат выходных данных

В случае, если дверь упрётся в стену и не заденет шкаф, выведите «Yes». В противном случае выведите «No».

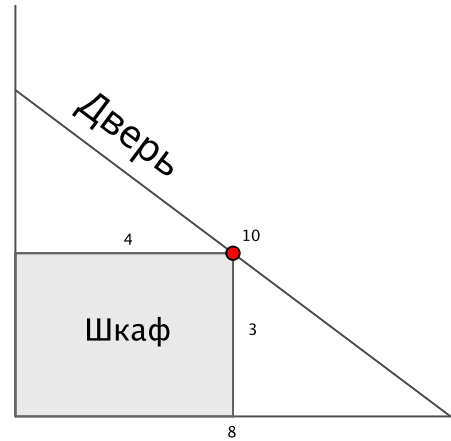
Примеры

стандартный ввод	стандартный вывод
2 2 6 4	Yes
4 3 10 8	No
1 1 1 2	No
1 1 1 3	Yes

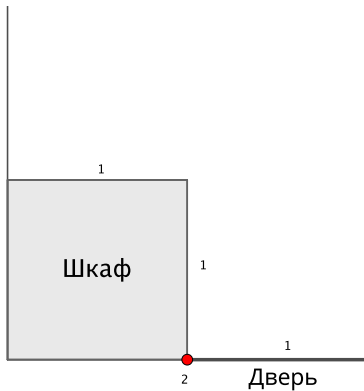
Замечание



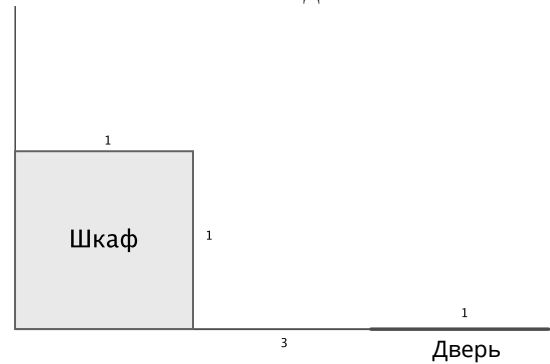
В первом примере дверь упрётся в дальнюю от неё стену и не заденет шкаф.



Во втором примере дверь упрётся в стену, при этом она коснётся шкафа, поэтому считается, что она его задела.



В третьем примере дверь упрётся в ближнюю стену, при этом её конец будет касаться шкафа, поэтому считается, что она его задела.



В четвёртом примере дверь упрётся в ближнюю стену, при этом её конец не будет касаться шкафа, поэтому считается, что она его не задела.