

Задача А. Антипалиндром

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Палиндром — это слово, которое читается одинаково в обе стороны. Значит, антипалиндром — слово, у которого i -й с начала и i -й с конца символы не совпадают ни для какого i . Например, слово “computer” является антипалиндромом, а слово “information” — нет.

Назовем число антипалиндромным, если его десятичная запись является антипалиндромной. По заданному числу x найдите минимальное антипалиндромное число, строго большее x .

Формат входных данных

Первая строка содержит целое число T ($1 \leq T \leq 1000$) — количество тестов. Каждый тест состоит из единственного числа x ($1 \leq x \leq 10^{100}$), записанного на отдельной строке.

Формат выходных данных

Для каждого числа выведите в отдельную строку минимальное большее его антипалиндромное число.

Пример

стандартный ввод	стандартный вывод
3	10
6	20
19	1010
98	

Задача В. Пятнашки

Ограничение по времени: 0.5 секунд
Ограничение по памяти: 64 мегабайта

Дана квадратная таблица размера 4×4 . В клетках записаны целые числа от 1 до 15, а также одна пустая клетка.

За один ход разрешается переместить в пустую клетку число, находящееся в соседней клетке. Две клетки считаются соседними, если они имеют общую сторону.

Цель игры — за минимальное число ходов привести таблицу к конечной позиции, в которой числа упорядочены по строкам по возрастанию, а пустая клетка находится в правом нижнем углу, то есть

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	0

Необходимо написать программу, которая по заданной начальной позиции находит минимальное количество ходов, с помощью которых можно достичь конечной позиции.

Формат входных данных

Вводятся 16 целых чисел, записанных по строкам таблицы. Число 0 обозначает пустую клетку.

Формат выходных данных

В первой строке выведите минимальное количество ходов, необходимое для достижения конечной позиции.

Во второй строке выведите последовательность чисел, которые перемещаются в пустую клетку на каждом ходе.

Гарантируется, что конечная позиция достижима, и минимальное число ходов меньше 25.

Пример

стандартный ввод	стандартный вывод
1 2 3 4 5 0 6 8 9 10 7 11 13 14 15 12	4 6 7 11 12

Задача С. Привлекательные участки

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вы купили участок земли длиной K километров. Он настолько узкий, что для целей задачи его можно рассматривать как ломаную линию, идущую с запада на восток, меняющую высоту.

Известны высоты участка (в метрах) в $K + 1$ равномерно расположенных измерительных точках M_0, M_1, \dots, M_K . Точка M_i находится на расстоянии i километров от западного конца участка.

Границы между соседними земельными участками обозначаются деревянными столбами. Столбы можно устанавливать только в измерительных точках, причём в каждой точке может быть не более одного столба.

Сейчас уже установлены два столба: в точке 0 км и в точке K км. Измерительная точка со столбом считается принадлежащей сразу двум соседним участкам, поэтому ваш участок включает все точки от 0 до K включительно.

Участок считается привлекательным, если внутри него существуют три измерительные точки такие, что западная и восточная из этих трёх точек обе строго выше третьей точки, либо обе строго ниже неё.

Иными словами, существуют индексы $i < j < k$ внутри участка такие, что выполняется одно из условий:

$$M_i > M_j \text{ и } M_k > M_j,$$

или

$$M_i < M_j \text{ и } M_k < M_j.$$

Заметим, что эти три точки не обязаны быть соседними, и крайние точки тройки не обязательно совпадают с границами участка.

Ваш исходный участок уже является привлекательным, но вы хотите увеличить его ценность, разделив его на несколько меньших участков. Вы можете добавить дополнительные столбы, чтобы разбить исходный участок на несколько частей, причём каждая получившаяся часть должна быть привлекательной.

Требуется определить максимальное количество дополнительных столбов, которые можно установить.

Формат входных данных

В первой строке задано число тестов T ($1 \leq T \leq 100$).

Далее следуют T тестовых случаев. Каждый тест начинается с целого числа K ($4 \leq K \leq 10000$) — длины участка. В следующей строке заданы $K + 1$ целых чисел M_0, M_1, \dots, M_K , где M_i — высота в точке i км от западного конца ($0 \leq M_i \leq 10000$). Гарантируется, что исходный участок является привлекательным.

Формат выходных данных

Для каждого теста выведите одно число — максимальное число столбов, которые можно добавить.

Пример

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

Задача D. Исследовательский консорциум

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Исследовательский консорциум построил новую систему базы данных для своего дата-центра. База данных состоит из одного главного компьютера (мастера) и N рабочих компьютеров. Рабочие компьютеры имеют идентификаторы от 0 до $N - 1$. Каждый рабочий компьютер хранит ровно один бит информации.

Мастер получает строку длины N , состоящую из нулей и единиц, и отправляет i -й бит i -му рабочему компьютеру для хранения. После этого мастер считывает биты обратно с рабочих компьютеров и возвращает их пользователю в том же порядке.

При нормальной работе система должна возвращать ту же строку, что и получила. Однако B рабочих компьютеров сломаны. Сломанные компьютеры способны правильно сохранять полученные биты, но при попытке чтения мастер не получает от них никакого ответа. В результате операция возвращает строку длины $N - B$, содержащую только биты, считанные с исправных компьютеров, в порядке возрастания их идентификаторов.

Например, пусть, $N = 5$, а рабочие компьютеры с идентификаторами 0 и 3 сломаны ($B = 2$).

- Для строки 01101 операция возвращает 111;
- Для строки 00110 операция возвращает 010;
- Для строки 01010 операция возвращает 100;
- Для строки 11010 операция также возвращает 100.

Из-за того, что база данных находится в защищённом подземном хранилище, каждый вызов операции занимает очень много времени. Вам необходимо определить, какие именно рабочие компьютеры сломаны, используя не более 5 вызовов.

Протокол взаимодействия

Это интерактивная задача.

В начале программа получает одно целое число T ($1 \leq T \leq 100$) — количество тестовых случаев.

Далее следуют T тестов. Для каждого теста сначала задаются два целых числа N и B ($2 \leq N \leq 1024$, $1 \leq B \leq \min(15, n - 1)$) — количество рабочих компьютеров и количество сломанных компьютеров.

После этого ваша программа может вывести не более 5 строк. Каждая строка должна состоять ровно из N символов 0 и 1. В ответ жюри выводит строку длины $N - B$, соответствующую результату выполнения операции.

Когда ваша программа определит все B сломанных компьютеров, она должна вывести B различных целых чисел — их идентификаторы в отсортированном порядке. Этот вывод не считается запросом. Если программа превысит лимит запросов, жюри выведет -1 и завершит взаимодействие. Если ответ неверен, жюри выведет -1. Если ответ верен, жюри выведет 1 и начнётся следующий тест (или взаимодействие завершится).

Пример

стандартный ввод	стандартный вывод
1	
5 2	
111	01101
010	00110
100	01010
100	11010
100	0 3
1	

Задача Е. Сортировка колоды

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

У вас появилась новая колода карт. Каждая карта имеет ранг — целое число от 1 до R , и масть — целое число от 1 до S . Для каждой пары (r, s) существует ровно одна карта, поэтому всего в колоде $R \cdot S$ карт. Будем обозначать карту с рангом r и мастью s как (r, s) .

Изначально новая колода отсортирована сверху вниз по масти в порядке возрастания, а при равенстве мастей — по рангу в порядке возрастания. То есть сначала идут карты

$$(1, 1), (2, 1), \dots, (R, 1),$$

затем

$$(1, 2), (2, 2), \dots, (R, 2),$$

и так далее до (R, S) .

Ваша цель — переупорядочить колоду так, чтобы она была отсортирована по рангу. Это означает, что все карты одного ранга должны идти подряд, а ранги должны располагаться по возрастанию. Порядок мастей внутри одного ранга не имеет значения.

Например, при $R = 4$ и $S = 2$ допустимым итоговым порядком может быть:

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

Вы хотите выполнить сортировку, используя только следующую операцию:

Сначала вы берёте одну или более карт с вершины колоды и откладываете их в стопку A . Затем вы берёте одну или более следующих карт с вершины колоды и откладываете их в стопку B . После этого вы кладёте стопку A обратно на колоду, а затем поверх неё кладёте стопку B . Таким образом, два верхних подряд идущих блока карт меняются местами, а остальные карты в колоде не затрагиваются.

По заданным R и S требуется найти последовательность таких операций, которая приведёт колоду к сортировке по рангу и использует минимально возможное число операций.

Формат входных данных

В первой строке задано число тестовых случаев T ($1 \leq T \leq 100$). Далее следуют T строк, каждая из которых содержит два целых числа R и S ($2 \leq R, S \leq 40$) — количество рангов и количество мастей в колоде.

Формат выходных данных

Для каждого тестового случая выведите в первой строке число k — минимальное число операций.

Затем выведите k строк, описывающих операции. В i -й строке должны быть два числа a_i и b_i , означающие, что в i -й операции вы берёте a_i карт в стопку A , а затем b_i карт в стопку B .

Пример

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

Задача F. Вопросы в зоопарке

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В зоопарке находятся n животных, пронумерованных от 1 до n . Каждое животное — кролик или кошка. Их рост попарно различен. Лис не умеет отличать кроликов от кошек. Поэтому он задал каждому животному вопрос: “Сколько животных того же вида, что и ты, выше тебя ростом?”

Каждое животное знает относительный порядок ростов и отвечает правдиво: кролик считает только кроликов, кошка — только кошек. По данным ответам требуется определить, сколькими различными способами можно назначить каждому животному его вид (кролик или кошка), так чтобы все ответы были корректными.

Два способа назначения считаются различными, если существует хотя бы одно животное, которое является кроликом в одном случае и кошкой в другом.

Формат входных данных

Первая строка содержит целое число n ($1 \leq n \leq 3 \cdot 10^5$). Вторая строка содержит n целых чисел a_i ($0 \leq a_i \leq 3 \cdot 10^5$) — ответ i -го животного.

Формат выходных данных

Выведите одно число — ответ на задачу по модулю 998244353.

Примеры

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

Задача G. Варвары

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Племя варваров окружило страну. Граница страны имеет форму строго выпуклого многоугольника и задана вершинами $(bx[i], by[i])$, перечисленными в порядке против часовой стрелки.

Внутри страны расположены N городов, которые мы будем считать точками $(cx[i], cy[i])$.

Варвары равномерно распределились вдоль всей границы страны так, что расстояние между соседними варварами (измеряемое вдоль периметра границы) одинаково. Известно, что один из варваров находится точно в вершине $(bx[0], by[0])$. Общее число варваров равно

$$31179239! = 1 \cdot 2 \cdot \dots \cdot 31179239.$$

Варвары хотят разделиться на N равных по численности групп так, чтобы каждая группа атаковала ровно один город. На формирование групп нет никаких ограничений — в частности, участники одной группы не обязаны стоять рядом на границе.

Известно, что варвар проходит расстояние в 1 единицу за 1 час. Считайте, что варвары и города являются точками на плоскости. Определите минимальное время (в часах), через которое последний варвар из последней группы сможет добраться до соответствующего города, если все варвары движутся оптимальным образом.

Формат входных данных

В первой строке задано число M ($3 \leq M \leq 50$) — количество вершин многоугольника.

В следующих M строках заданы координаты вершин: по два целых числа bx_i, by_i ($-1000 \leq bx_i, by_i \leq 1000$).

В следующей строке задано число N ($1 \leq N \leq 5$) — количество целей.

В следующих N строках заданы координаты целей: по два целых числа cx_j, cy_j ($-1000 \leq cx_j, cy_j \leq 1000$).

Гарантируется, что: многоугольник является строго выпуклым; вершины заданы в порядке обхода против часовой стрелки; все цели различны; все цели лежат строго внутри многоугольника.

Формат выходных данных

Выведите одно вещественное число — минимальное время в часах.

Ответ будет принят, если абсолютная или относительная ошибка не превосходит 10^{-9} .

Примеры

стандартный ввод	стандартный вывод
4 0 0 2 0 2 2 0 2 1 1 1	1.414213562378493
4 0 0 3 0 3 3 0 3 2 1 2 2 1	2.236067977634896

Замечание

Многоугольник называется строго выпуклым, если он выпуклый и никакие три его последовательные вершины не лежат на одной прямой.

Задача Н. Локализация Квантового Исследователя

Ограничение по времени: 2 секунды
Ограничение по памяти: 1024 мегабайта

Вы являетесь частью Отдела Квантовых Исследований и запускаете свой первый автономный зонд, Кванта, предназначенный для исследования плоских, неизученных поверхностей на далеких планетах.

Кванта должна ориентироваться на этих поверхностях, даже если она потеряет доступ к своим основным датчикам местоположения. Зонд полагается на специальные квантовые маяки, размещенные на местности для резервной навигации.

Формально, местность представляет собой декартову плоскость (координаты в метрах). Существует N квантовых маяков в различных позициях. Кванта может обнаруживать все маяки в пределах манхэттенского расстояния (также называемого L_1 расстоянием) не более чем D метров от своего текущего положения. Однако зонд воспринимает только относительные местоположения этих маяков, а не идентифицирует их.

Например, если Кванта обнаруживает маяк на 3 метра на север и 4 метра на запад, она знает только, что есть маяк на этом относительном смещении, а не какой это конкретно маяк.

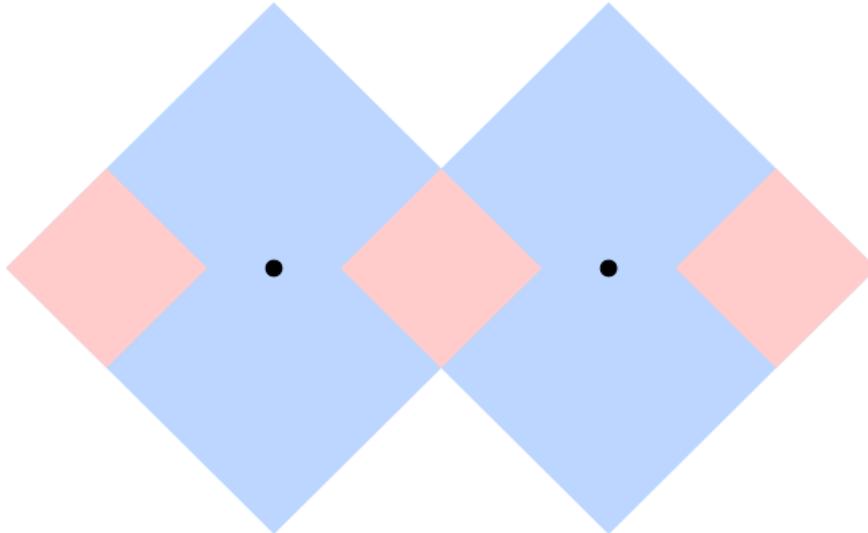
Пусть относительная информация, полученная Квантой в позиции (x, y) , будет

$$\text{Info}(x, y) := \{(z - x, w - y) \mid (z, w) \text{ — это маяк и } |z - x| + |w - y| \leq D\}.$$

Точка (x_1, y_1) называется неразличимой, если существует другая точка (x_2, y_2) такая, что

$$\text{Info}(x_1, y_1) = \text{Info}(x_2, y_2).$$

Все остальные точки называются различимыми.



Кванта будет развернута в точке, выбранной равномерно случайным образом из всех местоположений в пределах D метров (по манхэттенскому расстоянию) от как минимум одного маяка (точки, где $\text{Info}(x, y)$ не пусто). Вероятность развертывания в непрерывной области пропорциональна её площади в квадратных метрах.

Учитывая позиции всех квантовых маяков, определите вероятность того, что Кванта развернута в различимом местоположении. Выведите эту вероятность в виде сокращенной дроби y/z .

Формат входных данных

Первая строка содержит два целых числа N и D ($2 \leq N \leq 2000$, $1 \leq D \leq 10^7$) — количество маяков и максимальное расстояние обнаружения. Каждая из следующих N строк содержит два целых числа X_i и Y_i ($-10^9 \leq X_i, Y_i \leq 10^9$) — координаты i -го маяка. Все позиции маяков различны.

Формат выходных данных

Выведите строку, содержащую два целых числа y и z — вероятность того, что Кванта находится в различимом местоположении, представленная в виде сокращенной дроби y/z .

Примеры

стандартный ввод	стандартный вывод
2 4 0 0 5 0	27 119
2 1 0 0 5 0	0 1
2 4 0 0 4 4	0 1
2 4 0 0 5 1	1 5
3 4 0 0 1 1 2 3	101 109

Замечание

В первом примере диапазон обнаружения составляет $D = 4$, и есть два маяка с координатами $(0, 0)$ и $(5, 0)$.

Точки, где Кванта может обнаружить только один маяк в диапазоне (синие области на рисунке), являются неразличимыми, потому что относительная информация одинакова, если Кванта находится рядом с другим маяком. Точки, где Кванта может обнаружить оба маяка (красная средняя область), являются различимыми, потому что полученная информация уникальна.

Общая площадь различимых областей составляет 27 квадратных метров, а общая площадь всех областей в пределах D от как минимум одного маяка составляет 119 квадратных метров. Следовательно, вероятность разветвления в различимом местоположении составляет $27/119$.

Задача I. Объекты в ориентированном дереве

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дан ориентированный ациклический граф в виде дерева с n вершинами, пронумерованными от 1 до n , и $n - 1$ ребрами, направленными к вершине 1.

Для каждой вершины $i > 1$ задано:

- p_i — вершина, в которую ведет единственный исходящий путь из i ;
- c_i — количество объектов, находящихся в i на момент $t = 0$;
- m_i — максимальное количество объектов, которое может переместиться из i в p_i за один шаг времени.

Вершина 1 не имеет ограничений на приём объектов.

В каждый момент времени каждый объект может либо остаться в своей вершине, либо пройти через один или несколько путей к вершине 1, при этом ограничение m_i на каждом пути должно соблюдаться. То есть за один шаг времени через путь из вершины i в p_i может пройти не более m_i объектов.

Дан массив t_1, t_2, \dots, t_k . Для каждого t_i требуется определить максимальное количество объектов, которые могут оказаться в вершине 1 к моменту времени t_i , если перемещения оптимальны.

Формат входных данных

Первая строка содержит два целых числа n и k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^5$).

Следующие $n - 1$ строк описывают вершины $i = 2 \dots n$. В каждой строке три целых числа: p_i ($1 \leq p_i < i$), c_i ($1 \leq c_i \leq 10^9$), m_i ($0 \leq m_i \leq 10^9$).

Следующие k строк содержат по одному числу: t_i ($1 \leq t_i \leq 10^9$).

Формат выходных данных

Выведите k строк. В строке i выведите максимальное количество объектов, которые могут оказаться в вершине 1 к моменту времени t_i .

Пример

стандартный ввод	стандартный вывод
4 5	25
1 1 5	10
2 12 7	5
3 12 3	15
5	20
2	
1	
3	
4	

Задача J. Без воды в условии

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Найдите все x , такие что $x^k \bmod p = 1$.

Формат входных данных

Первая строка содержит два целых числа: p и k ($2 \leq p \leq 10^9$, p — простое, $2 \leq k \leq p - 1$).

Формат выходных данных

Выведите количество найденных x , а во второй строке — сами x в возрастающем порядке. Гарантируется, что количество чисел не будет превышать 10000.

Пример

стандартный ввод	стандартный вывод
5 2	2 1 4

Задача К. Игра с диаметром

Ограничение по времени: 1 секунда
Ограничение по памяти: 128 мегабайт

В физтех-школе прикладной математики и информатики (ФПМИ) проходит необычная стратегическая игра. Дано дерево из N вершин. Играют два студента ФПМИ. В начале партии они по очереди выбирают вершины. Первым ходит первый игрок, затем второй. За один ход игрок выбирает ровно одну вершину, которая ещё не была выбрана. Игра продолжается до тех пор, пока все вершины не окажутся выбранными.

После окончания игры рассмотрим вершины первого игрока. Разброс вершин первого игрока определяется как максимальное расстояние между двумя его вершинами. Расстоянием между двумя вершинами считается число ребер на кратчайшем пути между ними.

Первый игрок стремится минимизировать этот разброс, а второй игрок стремится сделать его как можно больше. Требуется определить значение разброса первого игрока при оптимальной игре обоих игроков.

Формат входных данных

В первой строке задано целое число N ($3 \leq N \leq 10^5$) — количество вершин.

В следующих $N - 1$ строках заданы два целых числа u и v , означающие, что вершины u и v соединены неориентированным ребром. Гарантируется, что граф является деревом.

Формат выходных данных

Выведите одно число — разброс первого игрока при оптимальной игре обоих игроков.

Примеры

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

Задача L. Соревнования по робототехнике

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вы построили робота для соревнования по робототехнике. Одна из задач, которую должен решить робот, заключается в подсчёте количества кубиков в квадратной сетке.

Сетка состоит из N строк и N столбцов. В каждой ячейке находится стопка из одинаковых кубиков, количество которых — целое число от 1 до 5000 включительно.

К сожалению, робот плохо умеет собирать информацию. Единственное, что он может определить, — это высоту самой высокой стопки в каждой строке и в каждом столбце.

Вы решили компенсировать это, написав умную программу. По информации, собранной роботом, требуется определить минимально возможное и максимально возможное общее количество кубиков в сетке.

Формат входных данных

В первой строке задано целое число N ($3 \leq N \leq 5000$) — размер сетки. Во второй строке задано N целых чисел r_i ($1 \leq r_i \leq 5000$) — высота самой высокой стопки в строке i . В третьей строке задано N целых чисел c_i ($1 \leq c_i \leq 5000$) — высота самой высокой стопки в столбце i .

Гарантируется, что входные данные непротиворечивы, то есть всегда существует хотя бы одна расстановка кубиков, соответствующая заданным значениям.

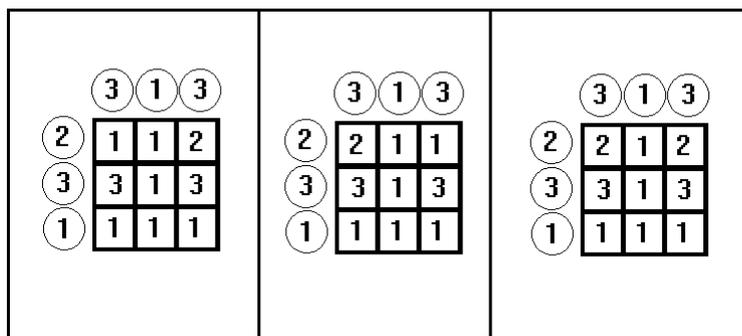
Формат выходных данных

Выведите два целых числа: минимально возможное и максимально возможное общее количество кубиков в сетке.

Пример

стандартный ввод	стандартный вывод
3 2 3 1 3 1 3	14 15

Замечание



На рисунке показаны все возможные конфигурации для первого примера. Первые две из них содержат по 14 кубиков, а третья — 15 кубиков. Следовательно, минимально возможное общее количество кубиков равно 14, а максимально возможное — 15.

Задача М. Ксорцистка

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Ксорцистка обладает необычной способностью — она умеет невероятно быстро сортировать списки. Глядя на список неотрицательных целых чисел a_1, a_2, \dots, a_N , она мгновенно находит такое неотрицательное целое число X , что

$$a_1 \oplus X \leq a_2 \oplus X \leq \dots \leq a_N \oplus X,$$

где \oplus обозначает побитовую операцию XOR.

После этого Ксорцистке достаточно заменить каждое число a_i на $a_i \oplus X$, и список оказывается отсортированным.

Многие компании пользовались её услугами для сортировки огромных массивов данных. Однако однажды Ксорцистка с ужасом обнаружила, что утратила свою способность. Теперь ваша задача — написать программу, которая позволит ей сохранить работу.

Вам дан список неотрицательных целых чисел a_1, a_2, \dots, a_N , а также Q изменений. Каждое изменение задаётся парой (p_i, v_i) и означает, что значение a_{p_i} заменяется на v_i .

Требуется вывести $Q + 1$ целых чисел c_0, c_1, \dots, c_Q , где c_i — это минимальное неотрицательное целое число X , при котором последовательность

$$a_1 \oplus X, a_2 \oplus X, \dots, a_N \oplus X$$

является неубывающей после применения первых i изменений. Если такого числа X не существует, выведите -1 .

Формат входных данных

В первой строке задано целое число N ($1 \leq N \leq 10^6$) — количество элементов в списке.

Во второй строке записаны N неотрицательных целых чисел a_1, a_2, \dots, a_N ($0 \leq a_i < 2^{30}$).

В третьей строке задано целое число Q ($0 \leq Q \leq 10^6$) — количество изменений.

В следующих Q строках заданы по два целых числа p_i и v_i ($1 \leq p_i \leq N$, $0 \leq v_i < 2^{30}$), обозначающие замену a_{p_i} на v_i .

Формат выходных данных

Выведите $Q + 1$ строку. В i -й строке должно быть выведено число c_i — минимальное возможное значение X , при котором список отсортирован после выполнения первых i изменений, либо -1 , если такого значения не существует.

Пример

стандартный ввод	стандартный вывод
3	0
0 1 4	2
3	-1
2 7	4
3 3	
1 4	