

Задача А. Новая игра

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

Бродя по различным местам Фарлума, Хорнет нашла странного жука. Он обещал помочь Хорнет, но только в том случае, если она обыграет его в странной игре.

Игра происходит по следующим правилам:

- сначала жук дает ей два целых положительных числа a и b ;
- после чего спрашивает у Хорнет, существует ли целое неотрицательное число c , для которого $a + c$ является подстрокой $b + c$.

Здесь под «число x является подстрокой числа y » подразумевается, что запись числа x в десятичной системе совпадает с некоторым отрезком записи числа y в десятичной системе счисления.

Так как Хорнет очень хочет выиграть, вы должны помочь ей с этим. А именно: выведите подходящее число c , либо определите, что такого числа не существует.

Формат входных данных

В первой строке дано целое число a ($1 \leq a \leq 10^{10\,000}$).

Во второй строке дано целое число b ($1 \leq b \leq 10^{10\,000}$).

Формат выходных данных

Если существует подходящее c , длина которого не превосходит 20 000, выведите его. Если вариантов ответа несколько, разрешается вывести любой из них.

Если же подходящего c не существует, выведите число -1 .

Примеры

стандартный ввод	стандартный вывод
5 90	4
1234 2345	-1
12 14453212	2

Задача В. Торговцы

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

Путешествуя по Фарлуму, Хорнет встретила n торговцев. Каждый из них продавал ровно один уникальный товар, и изначально цена товара у i -го торговца была равна a_i .

Хорнет интересуется, сколько суммарно придется потратить денег, чтобы купить несколько подряд идущих товаров. Есть только одна проблема: торговцы постоянно меняют цену на свой товар. Формально, q раз происходят события одного из трех типов:

- «1 i x » — i -й торговец решает сделать цену своего товара равной x ;
- «2 l r » — торговцы с l -го по r -й включительно *инвертируют* цены своих товаров;
- «3 l r » — Хорнет решает заглянуть ко всем торговцам с l -го по r -го включительно, и хочет посчитать, в какую суммарную стоимость обойдутся ей их товары.

Под *инвертированием* цены подразумевается следующий процесс:

1. торговец записывает текущую цену в двоичной системе счисления с k знаками (гарантируется, что k бит достаточно);
2. если двоичная запись получилась длины меньше k , торговец дополняет ее ведущими нулями;
3. после чего торговец инвертирует все биты данного числа (например, из «01001» в «10110»);
4. и переводит полученную двоичную запись обратно в десятичную систему счисления.

Вам необходимо помочь Хорнет и ответить на все ее запросы.

Формат входных данных

В первой строке даны два целых числа n и k — число торговцев и размер всех цен в двоичной системе, соответственно ($1 \leq n \leq 2 \cdot 10^5$; $1 \leq k \leq 25$).

Во второй строке даны n чисел a_i — начальная цена товаров у торговцев ($0 \leq a_i < 2^k$).

В третьей строке дано единственное число q — количество запросов ($1 \leq q \leq 2 \cdot 10^5$).

В следующих q строках даны описания запросов в формате, данном в условии: сначала тип запроса, затем параметры ($1 \leq i, l, r \leq n$; $0 \leq x < 2^k$).

Формат выходных данных

Для каждого запроса третьего типа (узнать сумму), выведите ответ на него в отдельной строке.

Пример

стандартный ввод	стандартный вывод
5 3	20
1 2 3 4 5	7
7	0
1 2 7	22
3 1 5	
3 2 2	
2 2 2	
3 2 2	
2 1 5	
3 1 5	

Задача С. Битвы с боссами

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

У Хорнет есть n боевых характеристик, уровень каждой описывается целым числом a_i . Всего за долгое время своих приключений она встретила m боссов, и для каждого из них известны их значения тех же характеристик, тоже по n у каждого: j -я характеристика у i -го босса равна $b_{i,j}$.

Хорнет сражалась с боссами не обязательно в порядке увеличения их номеров и не обязательно по одному разу. У нее есть массив номеров боссов s длины k , по которому определяется, с кем она сражалась. Числа в массиве s могут повторяться, и не обязательно каждое число от 1 до m в нем встречается.

Известно, что в процессе приключений могли происходить два вида событий:

1. «1 i x » — значение i -й характеристики Хорнет изменилось на x (иными словами, к a_i добавилось x);
2. «2 j s » — Хорнет сразилась с боссом под номером c_j , и победила, если $s = \text{«win»}$, или проиграла, если $s = \text{«loss»}$.

Хорнет побеждает c_j -го босса тогда и только тогда, когда все ее характеристики имеют уровень не меньше, чем у него, то есть когда для любого i верно $a_i \geq b_{c_j,i}$. Соответственно, для поражения достаточно, чтобы хотя бы одна из характеристик Хорнет была ниже, чем у босса.

К сожалению, значения c_j были потеряны, но сохранились записи обо всех событиях. Определите, существует ли такой набор c_j , при котором все исходы боев будут совпадать с исходами в записях, или же в записях есть ошибка.

Формат входных данных

В первой строке даны три целых числа n , m и k — количество боевых характеристик, число боссов и число событий ($1 \leq n, m, k \leq 1000$).

Во второй строке перечислены n целых чисел a_i — изначальные уровни характеристик Хорнет ($0 \leq a_i \leq 500$).

В следующих m строках в том же формате даны характеристики боссов по n чисел $b_{i,j}$ в каждой строке ($0 \leq b_{i,j} \leq 500$).

В следующих k строках даны записи о событиях в формате, данном в условии ($1 \leq i \leq n$; $1 \leq j \leq m$; $1 \leq x \leq 500$; $s \in \{\text{«win»}, \text{«loss»}\}$).

Формат выходных данных

Выведите «Yes» (без кавычек), если существует такой порядок сражений с боссами s , при котором записи корректны, и «No» иначе.

Обратите внимание, что не требуется, чтобы Хорнет сразилась с каждым боссом хотя бы один раз.

Примеры

стандартный ввод	стандартный вывод
3 2 4 0 0 0 1 2 0 0 2 1 1 1 2 1 2 3 2 3 loss 2 2 win	Yes
3 2 5 0 0 0 1 0 2 0 2 1 1 1 2 1 2 3 2 2 win 1 3 3 2 3 loss	No

Замечание

В первом примере подходит массив $c = [*, 1, 2, *]$ (под $*$ подразумевается любое значение, так как про них в записях нет данных).

До битвы с боссами характеристики Хорнет будут равны $[2, 3, 0]$. С такими характеристиками она может победить первого босса, но проиграет второму.

Во втором примере нет ни одного подходящего массива, так как перед последним запросом её характеристики равны $[2, 3, 3]$, из-за чего она не может проиграть ни одному боссу.

Задача D. Скелеты, кости, кладбище, черепа

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

Фарлум — королевство шелковых троп. Каждая тропа — натянутая прямая нить, связывающая две деревни. Коварный злодей Тиран Череп собирается бить по паутине троп вертикальными ударами.

Представим королевство как граф: вершины — деревни с их координатами (x_i, y_i) на плоскости, а рёбра — нити-тропы между ними. Отрезки на плоскости, соответствующие дорогам, могут пересекаться вне деревень, но перейти с одной дороги на другую в таких точках пересечения нельзя.

У жителей королевства есть q предположений, куда Тиран Череп захочет нанести удар. Если злодей наносит удар по вертикальной прямой с координатой X , то удаляются:

- деревни, у которых $x_i = X$;
- дороги между деревнями i, j такие, что: $\min(x_i, x_j) \leq X \leq \max(x_i, x_j)$.

Помогите жителям, и для каждого возможного удара сообщите, сколько *компонент связности* останется в королевстве после этого удара. Все запросы независимы: при ответе на текущее предположение предыдущие учитывать не надо.

Напомним, что компонента связности в графе — это такое множество вершин, в котором из каждой вершины можно попасть по рёбрам в любую другую (и нельзя попасть в вершины, не принадлежащие этому множеству).

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t — количество наборов входных данных ($1 \leq t \leq 5000$). Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит три целых числа n, m, q — количество деревень, количество дорог и количество возможных ударов по паутине троп ($1 \leq n \leq 10^5$; $0 \leq m \leq \min(10^5, \frac{n(n-1)}{2})$; $1 \leq q \leq 10^5$).

Вторая строка каждого набора входных данных содержит n целых чисел x_i — x -координаты деревень. Третья строка в том же формате перечисляет y_i — y -координаты деревень ($-10^9 \leq x_i, y_i \leq 10^9$).

Следующие m строк содержат пары чисел u_i, v_i — номера деревень, соединённых дорогами в королевстве ($1 \leq u_i, v_i \leq n$; $u_i \neq v_i$).

В последней строке каждого набора входных данных содержатся q целых чисел — координаты возможных ударов, по модулю не превосходящие 10^9 .

Гарантируется, что никакие две деревни в одном наборе входных данных не совпадают, а также что три суммы n, m и q по всем наборам входных данных все не превосходят 10^5 .

Формат выходных данных

Для каждого набора входных данных выведите в соответствующей строке q чисел — ответы на запросы.

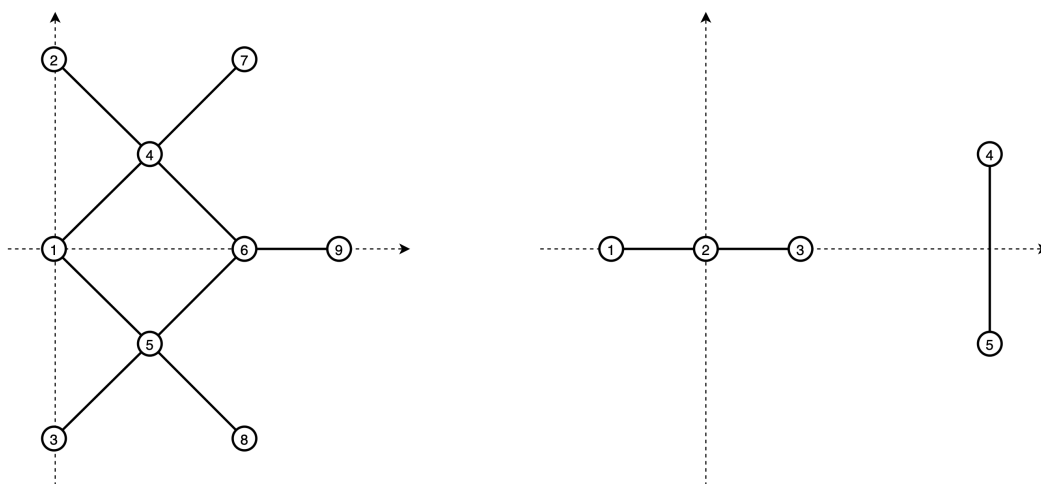
Обратите внимание, что запросы независимы.

Пример

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

Замечание

Рисунки графов к примерам из условия приведены ниже:



Задача Е. Путешествие по дереву

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

Во время своих приключений в Фарлуме, Хорнет m раз путешествовала по дереву скрытых проходов, состоящему из n вершин. В i -й раз она проходила по единственному пути от вершины a_i до вершины b_i .

Хорнет стало интересно, какое наибольшее число из этих путешествий можно выбрать так, чтобы каждое ребро дерева встречалось на выбранных путях в сумме не более одного раза.

Однако структура дерева стерлась из её памяти. Хорнет помнит только число вершин n и список из m пар (a_i, b_i) — начальных и конечных точек её путешествий. Поэтому она задумалась: насколько большим могло быть интересующее её число, если рассматривать все возможные деревья на n вершинах?

Иными словами, требуется найти максимальный размер подмножества данных пар вершин, для которого существует такое дерево на n вершинах, что каждое ребро дерева содержится в не более чем одном из путей выбранного множества.

Напомним, что дерево — это неориентированный связный граф без циклов. В дереве на n вершинах всегда ровно $n - 1$ ребро.

Формат входных данных

Первая строка содержит два целых числа n и m — количество вершин и количество путешествий Хорнет ($2 \leq n \leq 2 \cdot 10^5$, $1 \leq m \leq 2 \cdot 10^5$).

Следующие m строк содержат по два целых числа a_i и b_i ($1 \leq a_i, b_i \leq n$; $a_i \neq b_i$) — номера стартовой и конечной вершины очередного путешествия.

Формат выходных данных

Выведите одно целое число: максимально возможное число путешествий, которое можно выбрать среди данных так, чтобы существовало дерево на n вершинах, в котором никакое ребро не содержится в более чем одном путешествии.

Примеры

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

Задача F. Убежища

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	256 мегабайт

По одной туго натянутой шёлковой нити, пересекающей земли Фарлума, расселились n маленьких жителей. Каждый житель сидит на своей точке нити с координатой x_i . Всех жителей надо разместить по коконам на этой нити, которые будут служить им убежищами.

При размещении коконов на нити, каждый житель выдвинется к ближайшему к нему убежищу. Иными словами, если коконы расположатся на координатах $a_1, \dots, a_k$, то найдем для каждого x_i такое j_i , что $|x_i - a_{j_i}|$ минимально. Тогда *опасностью* этого набора коконов назовем $D = \max_{i=1}^n |x_i - a_{j_i}|$, то есть максимальное расстояние от жителя до ближайшего к нему кокона.

Хорнет получает q независимых поручений. В каждом поручении ей указывают:

- число k — сколько всего коконов-убежищ должно висеть на нити к концу работы;
- число r и набор $a_1, \dots, a_r$ — уже определённые положения для r из k убежищ.

От нее только требуется добавить ещё $k - r$ убежищ в любые точки нити (координаты не обязаны быть целыми) так, чтобы минимизировать итоговую *опасность* набора.

Для каждого поручения найдите минимально достижимое значение *опасности* итогового набора убежищ. Все поручения независимы.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t — количество наборов входных данных ($1 \leq t \leq 5000$). Далее следует описание наборов входных данных.

Первая строка каждого набора содержит два целых числа n и q — количество жителей и количество поручений ($1 \leq n, q \leq 35\,000$).

Вторая строка каждого набора содержит n целых чисел x_i — координаты жителей ($|x_i| \leq 10^8$).

Следующие q строк описывают поручения. В каждой строке записаны подряд целые числа k , r и $a_1, \dots, a_r$ для соответствующего поручения. Здесь k — общее число убежищ в итоге, r — число уже зафиксированных убежищ, а a_j — их координаты ($1 \leq k \leq 35\,000$; $0 \leq r \leq k$; $|a_j| \leq 10^8$).

Гарантируется, что сумма $n \cdot q$ по всем наборам входных данных не превосходит 35 000 и сумма r по всем наборам входных данных не превосходит 35 000.

Формат выходных данных

Для каждого набора входных данных выведите строку из q вещественных чисел — ответов на поручения. Ответ будет считаться правильным, если его абсолютная или относительная погрешность от ответа жюри не будет превышать 10^{-9} .

Пример

стандартный ввод	стандартный вывод
2	8 3.5 2 0 5 2.5
6 6	50.5 1 0.5 0 1
0 11 -5 4 -1 10	
1 0	
2 0	
3 0	
10 0	
2 1 0	
3 1 4	
4 5	
0 0 100 101	
1 0	
2 2 101 0	
3 1 50	
10 0	
2 1 100	

Замечание

В первом примере:

1. В первом запросе надо поставить только одно убежище, поэтому логично его ставить ровно посередине между крайними жителями (-5 и 11), и у каждого из них расстояние до убежища будет равно 8 .
2. Во втором запросе два убежища должны быть расположены посередине между -5 и 0 и между 4 и 11 .
3. В третьем запросе подходят убежища на позициях -3 , 2 и 10 .
4. В четвертом достаточно расположить по убежищу на позиции каждого жителя.
5. В пятом запросе второе убежище достаточно расположить около жителей на позициях 10 и 11 , тогда максимальное расстояние будет от жителя в точке -5 до убежища в 0 .
6. В шестом запросе надо добавить одно убежище около 10 и 11 , а второе в позиции -2.5 .

Задача G. Песнь Нити

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

На шёлковой равнине Фарлума расставлены n насекомых. У каждого из них одна из трёх ролей: ровно одна Певчая, $s - 1$ Эхо и $n - s$ Молчаливых.

В момент времени 0 единственная Певчая издаёт «клич». Звуковая волна не гаснет и распространяется равномерно во все стороны с одинаковой скоростью, то есть достигает точки на расстоянии d от Певчей за d времени. Затем каждый Эхо издаёт клич ровно в тот момент, когда до него впервые доходит чужой клич. Эта звуковая волна распространяется так же, как и первая. Разные звуковые волны не взаимодействуют друг с другом и не влияют друг на друга.

Слушатель находится в точке $(0, 0)$ и отмечает моменты прихода волн: сначала до него доходит ровно один клич, затем через некоторый ненулевой промежуток времени — одновременно ещё c_2 кличей, затем ещё через ненулевой промежуток — c_3 , и так всего k раз: получается последовательность $c_1, c_2, c_3, \dots, c_k$.

Обратите внимание, что каждый не-Молчаливый издаёт клич только один раз, поэтому $\sum_{i=1}^k c_i = s$.

Известны координаты всех n насекомых на плоскости. Требуется посчитать число способов выбрать из них ровно одну Певчую и ровно $s - 1$ Эхо (остальные автоматически становятся Молчаливыми) так, чтобы наблюдения Слушателя совпали с указанной последовательностью c_i . Ответ выведите по модулю $10^9 + 7$.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t — количество наборов входных данных ($1 \leq t \leq 500$). Далее следует описание наборов входных данных.

В первой строке набора входных данных даны целые числа n, k — число насекомых и ожидаемое число моментов, когда клич слышен в точке $(0, 0)$ ($1 \leq n \leq 1000$; $1 \leq k \leq \min(n, 10)$).

Во второй строке перечислены n целых чисел — x -координаты насекомых. В третьей строке также перечислены n целых чисел — y -координаты насекомых ($|x_i|, |y_i| \leq 10^9$).

В четвертой строке содержатся целые числа $c_1, \dots, c_k$ — в каком количестве кличи приходят в точку $(0, 0)$ в разные моменты времени ($1 \leq c_i \leq n$; $c_1 = 1$; $\sum_{i=1}^k c_i \leq n$). Значения c_i даны в хронологическом порядке (иными словами, их порядок совпадает с течением времени вперед).

Гарантируется, что сумма n по всем наборам входных данных не превосходит 1000.

Формат выходных данных

Для каждого набора входных данных выведите единственное число — количество способов назначить роли, по модулю $10^9 + 7$.

Пример

стандартный ввод	стандартный вывод
8	1
1 1	2
1	1
1	0
1	1
2 1	1
2 0	1
0 1	6
1	
2 2	
1 2	
0 0	
1 1	
3 2	
1 0 2	
0 1 0	
1 2	
3 2	
0 1 0	
0 0 1	
1 2	
4 3	
0 1 0 1	
0 0 1 1	
1 2 1	
6 3	
0 1 0 1 -1 1	
0 0 1 1 1 -1	
1 2 3	
3 2	
1 0 -1	
0 1 0	
1 1	

Замечание

Обратите внимание, что $c_1 = 1$ не гарантирует, что любое насекомое может стать Певчим.

Далее идут пояснения к примерам из условия. Будем считать, что звук идет со скоростью 1.

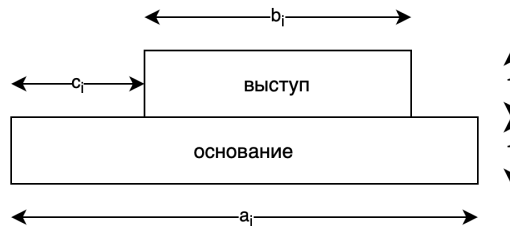
1. В первом тестовом наборе есть только одно насекомое в точке $(1, 1)$. Можно сделать его Певчим, тогда в момент времени $\sqrt{2}$ до точки слушателя дойдет один клич.
2. Во втором тестовом наборе есть две насекомых в точках $(2, 0)$ и $(0, 1)$. Можно сделать любую из них Певчей. В таком случае клич до слушателя дойдет в момент времени 2, если первую назначить Певчей, и в момент времени 1, если вторую назначить Певчей. Оба этих случая подходят под условие.
3. В третьем наборе координаты насекомых это $(1, 0)$ и $(2, 0)$. Если назначить второго Эхом, оба клича дойдут до Слушателя в одно и то же время, что не подходит под условие задачи.

Задача Н. Головоломка отрезков

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

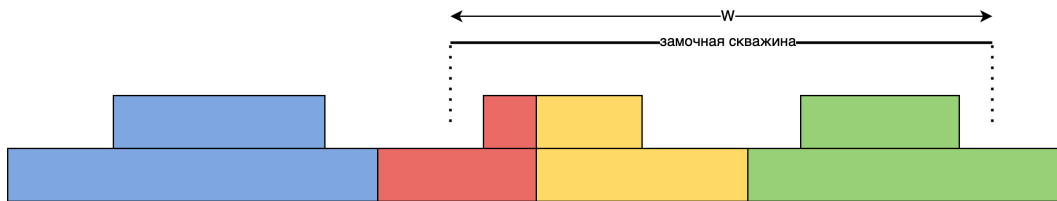
Иногда Хорнет приходится решать различные головоломки, появляющиеся у нее на пути.

Например, чтобы сразиться с Фантомом, ей сначала необходимо открыть дверь, ведущую к нему. В двери расположена замочная скважина ширины W , а также у Хорнет есть n ключей, i -й из которых имеет основание в виде прямоугольника размером $a_i \times 1$ и выступ в виде прямоугольника размером $b_i \times 1$ на расстоянии c_i от левого края основания. Гарантируется, что $a_i \geq b_i + c_i$, рисунок ключа ниже:



Ключи можно расположить подряд, совместив их основания, в любом порядке. Также любой ключ можно развернуть (заменить на симметричный с $c' = a_i - b_i - c_i$). Чтобы Хорнет могла пройти дальше, ей необходимо поместить выступы как можно большего числа ключей в скважину. Ключи считаются помещающимися в скважину, если расстояние между их крайними выступами не превосходит W .

На рисунке ниже в скважину получится поместить три ключа.



Помогите Хорнет решить эту головоломку!

Формат входных данных

В первой строке ввода даны два целых числа n и W — количество ключей и длина замочной скважины ($1 \leq n \leq 2 \cdot 10^5$; $1 \leq W \leq 10^9$).

В i -й из следующих n строк перечислены целые числа a_i , b_i и c_i — параметры i -го ключа ($1 \leq b_i \leq a_i \leq 10^9$; $0 \leq c_i \leq a_i - b_i$).

Формат выходных данных

В первой строке выведите k — максимальное количество ключей, которые получится вставить в замочную скважину.

В следующей строке выведите k целых ненулевых чисел от $-n$ до n , описывающих, какие ключи поместятся в скважину, в порядке слева направо. Если очередным в последовательности должен быть ключ номер i , выведите i , если этот ключ не надо разворачивать относительно его исходной конфигурации, и $-i$, если ключ надо заменить на симметричный.

Все использованные ключи должны быть различны.

Примеры

стандартный ввод	стандартный вывод
3 102 100 1 99 100 1 99 100 2 49	3 2 3 -1
4 30 10 2 1 10 2 6 11 10 1 11 10 0	4 -1 3 4 -2
4 10 3 1 2 6 3 2 7 4 2 4 2 2	3 4 1 -2

Задача I. Коллекция ягод

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

Хорнет решила выполнить задание по сбору ягод мха. Но чтобы ей не было скучно, она решила оценивать красоту каждой ягоды. Красота ягоды зависит от ее вида, который задаётся целым неотрицательным числом.

У Хорнет есть массив из n чисел a_i . Будем считать, что индексы массива идут с нуля. Когда Хорнет находит ягоду вида x , красота этой ягоды вычисляется следующим образом:

- рассмотрим двоичное представление числа x , записанное n битами: $\overline{b_{n-1}b_{n-2}\dots b_0}$, где b_0 — младший бит;
- перемножим все a_i , для которых $b_i = 1$; иными словами, рассмотрим $a_n^{b_n} \cdot a_{n-1}^{b_{n-1}} \cdot \dots \cdot a_1^{b_1}$;
- полученное значение обозначим за $f(x)$ и будем называть *красотой* ягоды.

Всего есть q мест, в которых можно найти ягоды мха, в i -м из них есть по одной ягоде видов от l_i до r_i включительно.

Помогите Хорнет найти суммарную красоту ягод в каждой из локаций, то есть вычислить $\sum_{j=l_i}^{r_i} f(j)$ для каждого i . Так как ответ может быть большим, выведите его остаток по модулю 998 244 353.

Формат входных данных

В первой строке дано целое число n — длина массива a , по которому Хорнет оценивает ягоды ($1 \leq n \leq 60$).

Во второй строке даны n целых чисел a_i — множители, участвующие в вычислении красоты ягод ($1 \leq a_i < 998\,244\,353$).

В третьей строке дано одно число q — количество локаций, в которых растут ягоды мха ($1 \leq q \leq 2 \cdot 10^5$).

В последующих q строках даны по два числа l_i и r_i , задающие отрезок видов ягод в i -й локации ($1 \leq l_i \leq r_i < 2^n$).

Формат выходных данных

Выведите q чисел, каждое на отдельной строке. В i -й строке следует вывести суммарную красоту всех ягод в i -й локации.

Так как ответы могут быть большими, выводите их по модулю 998 244 353.

Пример

стандартный ввод	стандартный вывод
4	2
2 3 5 7	24
4	575
1 1	76
2 5	
1 15	
5 9	

Задача J. Шёлковая лестница

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

Глубоко в заброшенных чертогах Фарлума Хорнет находит древний зал, где висит сеть из тончайших нитей. Когда-то здесь проходил ритуал «Шёлковой лестницы» — испытание, проверяющее терпение и точность.

В точках $(1, 1)$, $(2, 2)$, $\dots$, (n, n) закреплены колышки. От каждого из них тянутся две нити: одна вниз до оси Ox , другая влево до оси Oy . Вместе с самими осями Ox и Oy они образуют «лестницу» из вложенных квадратов с общим углом в $(0, 0)$.

Хорнет начинает путь в точке $(0, 0)$. Она может передвигаться в любую сторону

- вдоль оси Ox ,
- вдоль оси Oy ,
- вдоль нитей между колышком в (i, i) и точками $(i, 0)$ и $(0, i)$.

Задача Хорнет — посетить все n точек (i, i) для всех i от 1 до n за наименьшее число поворотов. Поворотом считается каждый раз, когда Хорнет меняет направление движения с горизонтального на вертикальное, с вертикального на горизонтальное или на 180° .

Помогите Хорнет и определите минимальное количество поворотов, необходимых для прохождения испытания.

Формат входных данных

В первой и единственной строке дано целое число n — количество колышков ($1 \leq n \leq 10^{12}$).

Формат выходных данных

В единственной строке выведите число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
1	1
4	10