

Задача А. Города

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

Вы играете в игру, в которой группа игроков по очереди произносит названия городов. Название города, которое вы произносите в свой ход, должно начинаться на ту же букву, на которую заканчивается название только что произнесенного города. При этом нельзя называть город, который уже звучал. Если нет подходящего города, вы выбываете из игры.

Учитывая последнее название города, сказанное перед вашим ходом, и список всех еще не использованных названий городов, сможете ли вы сделать ход? Если да, то сможете ли вы сделать так, чтобы игрок, ходящий за вами, выбыл из игры? Считайте, что он знает те же города, что и вы.

Формат входных данных

Первая строка ввода содержит одно слово — город, который только что назвал предыдущий игрок. Следующая строка содержит одно целое число n ($0 \leq n \leq 10^5$) — количество известных неиспользованных городов. Каждая из следующих n строк содержит одно название города.

Все названия городов (включая то, которое сказал предыдущий игрок) различные, состоят из строчных букв английского алфавита и имеют длину от 1 до 20.

Формат выходных данных

Если вы можете сыграть какое-либо название города, которое не оставляет ход следующему игроку, выведите первое такое название из списка, за которым выведите восклицательный знак.

В противном случае, если существует какое-либо название города, которое вы хотя бы можете сыграть на текущем ходу, выведите первое такое название. Иначе выведите вопросительный знак (в этом случае вам придется придумать вымышленный город в надежде, что остальные поверят вам).

Примеры

стандартный ввод	стандартный вывод
kirov 2 vologda arkhangelsk	vologda
moscow 2 dolgoprudniy novosibirsk	?
chelyabinsk 3 kursk volgograd kemerovo	kemerovo!

Задача В. Дерево в кучу

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

Вам дано подвешенное дерево, состоящее из n вершин. Вершина номер 1 является корнем. Каждая вершина также имеет значение v_i .

Вы хотите превратить это дерево в кучу. То есть вы хотите выбрать наибольшее возможное подмножество вершин, удовлетворяющих свойству кучи: для каждой пары вершин i, j в подмножестве, если вершина i является предком вершины j в дереве, то $v_i > v_j$. Обратите внимание, что равенство не допускается.

Определите максимальное число вершин, которые можно выбрать для формирования такого подмножества. Подмножество не обязательно должно образовывать поддереву.

Формат входных данных

Первая строка содержит целое число n ($1 \leq n \leq 2 \cdot 10^5$). Следующие n строк содержат описание вершин дерева. $i + 1$ -я строка содержит описание вершины i , состоящее из двух целых чисел v_i и p_i ($0 \leq v_i \leq 10^9$, $1 \leq p_i < i$). При этом $p_1 = 0$, что означает, что корнем является вершина с номером 1.

Формат выходных данных

Выведите одно число — ответ на задачу.

Примеры

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

Задача С. Кубок Физтеха по КНБ

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

Для начала расскажем правила игры «Камень-Ножницы-Бумага». Играют два игрока, они одновременно показывают рукой один из трех знаков – камень, ножницы или бумагу. Победитель определяется по следующим правилам:

- Бумага побеждает камень (бумага обёртывает камень).
- Камень побеждает ножницы (камень затупляет ножницы).
- Ножницы побеждают бумагу (ножницы разрезают бумагу).

Если игроки показали одинаковый знак, то засчитывается ничья. Матч состоит из нескольких таких раундов, и победителем объявляется игрок, выигравший большее количество раундов. Если оба игрока выиграли равное количество раундов, матч заканчивается ничьей.

Помимо вас, на кубок Физтеха по этой игре записались n участников. В турнире играют один на один, каждые два игрока играют k раундов друг против друга. Незадолго до начала кубка организаторы поняли, что играть придётся долго — надо сыграть $k \frac{n(n+1)}{2}$ раундов! Поэтому правила изменили, теперь каждый игрок должен придумать стратегию из k знаков. Теперь, вместо того чтобы играть по-настоящему, матч будет симулироваться на компьютере (две стратегии играют друг против друга).

Вы знаете стратегии всех n соперников, и теперь хотите придумать свою таким образом, чтобы сыграть все n матчей в ничью. Сколько различных таких стратегий существует?

Формат входных данных

Первая строка содержит два целых числа n и k ($1 \leq n \leq 100$, $1 \leq k \leq 20$). Следующие n строк содержат по k символов — стратегии участников. Камень обозначается R (Rock), ножницы S (Scissors), бумага P (Paper).

Формат выходных данных

Выведите одно число — ответ на задачу.

Пример

стандартный ввод	стандартный вывод
3 3 RPS SPR RRR	3

Задача D. Долгопрудный будущего

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

Конечно, в ближайшем будущем Москва поглотит Долгопрудный, но давайте помечтаем. Долгопрудный будущего состоит из улиц, параллельных сторонам света, то есть проходящих с запада на восток и с севера на юг. При этом город имеет идеальную топологию, и его идеализируют до прямоугольной сетки. Если вы хотите пройти от перекрестка $A = (A_x, A_y)$ до перекрестка $B = (B_x, B_y)$, кратчайший путь имеет длину $|A_x - B_x| + |A_y - B_y|$.

Однако такое уже есть в Манхэттене, потому в Долгопрудном пошли дальше. Они решили построить Диагональ — улицу, которая нарушит принципы манхэттенской метрики и введет свою, долгопрудную. Эта улица, единственная в городе, может не быть параллельна сторонам света. Таким образом, дорожная система будет состоять из дорог трех типов:

- Для каждого целого z существует дорога $x = z$.
- Для каждого целого z существует дорога $y = z$.
- Существует диагональ, описываемая уравнением $px + qy = r$.

Вы должны определять длину кратчайшего маршрута между перекрестками A и B , если двигаться можно только вдоль дорог и менять направление только на перекрестках.

Формат входных данных

Ввод содержит T тестовых случаев. Первая строка содержит целое число T ($1 \leq T \leq 60$).

Каждый тестовый случай описывается одной строкой, содержащей четыре целых числа A_x, A_y, B_x, B_y ($|A_x|, |A_y|, |B_x|, |B_y| \leq 10^6$) и три вещественных числа p, q, r ($|p|, |q| \leq 200, |r| \leq 10^9$).

Формат выходных данных

Для каждого тестового случая выведите ответ в отдельной строке. Ответ будет считаться правильным, если его абсолютная или относительная погрешность не превышает 10^{-9} .

Пример

стандартный ввод	стандартный вывод
2	10.000000000000
-1 5 3 -1 1.0 1.0 -5.2	3.414213562373
2 0 -1 1 1 1 1	

Задача Е. Окружности на столе

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

Есть прямоугольник, стороны которого параллельны осям координат, левый нижний угол имеет координаты $(-n, -m)$, правый верхний имеет координаты (n, m) . Вы хотите разместить в нем окружности таким образом, чтобы их центры находились в точках с целочисленными координатами, радиусы были целыми и положительными, а окружности не вылезали за края стола.

При этом должен быть способ упорядочить окружности таким образом, чтобы каждая окружность лежала внутри предыдущей и имела с ней не больше одной общей точки. Требуется посчитать число способов сделать такую конфигурацию окружностей. Два способа различны, если в одном из них есть окружность, которой нет в другом.

Формат входных данных

Даны два целых числа n и m ($1 \leq n, m \leq 1000$).

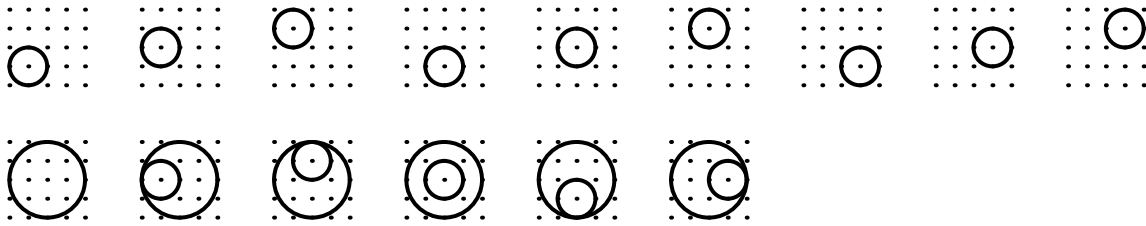
Формат выходных данных

Выведите одно число — ответ на задачу по модулю 998244353.

Пример

стандартный ввод	стандартный вывод
2 2	15

Замечание



Задача F. Массив и сумма степеней

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

Вам дан массив целых чисел a длины n и c монет. За одну монету вы можете увеличить одно число на один. Одно число можно увеличивать несколько раз. Вы не можете потратить больше, чем c монет.

Ваша задача максимизировать значение $\sum_i a_i^k$, при этом потратив как можно меньше монет. В этой задаче будем считать, что $0^0 = 0$.

Формат входных данных

В первой строке даны три целых числа n , c и k ($1 \leq n \leq 10^5$, $0 \leq c \leq 10^9$, $0 \leq k \leq 2$). Вторая строка содержит n целых чисел a_i ($|a_i| \leq 10^5$).

Формат выходных данных

Выведите два числа — максимальное значение $\sum_i a_i^k$ и минимальное число монет, необходимое, чтобы достичь такого значения суммы.

Примеры

стандартный ввод	стандартный вывод
3 2 2 -5 1 -4	50 2
2 0 0 0 0	0 0

Задача G. Конфеты по парам

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

У вас есть n конфет, у каждой есть тип (t_i) и ценность (c_i). Вы хотите выбрать $2k$ конфет так, чтобы их можно было разбить на k пар таким образом, чтобы конфеты в паре были разных типов. При этом вы хотите максимизировать суммарную ценность выбранных $2k$ конфет, а также понять, сколькими способами можно добиться максимальной суммарной ценности.

Формат входных данных

Первая строка содержит целое число k ($1 \leq k \leq \frac{n}{2}$). Вторая строка содержит целое число n ($2 \leq n \leq 1000$). Третья строка содержит n целых чисел t_i ($1 \leq t_i \leq 1000$). Четвертая строка содержит n целых чисел c_i ($1 \leq c_i \leq 10^7$).

Формат выходных данных

Если нельзя выбрать требуемым образом $2k$ конфет, выведите -1 . Иначе выведите два числа: максимальную суммарную ценность выбранных $2k$ конфет и число способов выбрать конфеты таким образом, по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
1 4 1 1 2 2 10 20 30 40	60 1
2 6 1 1 1 2 2 2 10 10 10 20 20 20	60 9

Задача Н. Выбор сеансов

Ограничение по времени: 2 секунды
Ограничение по памяти: 128 мегабайт

В прокате идет m фильмов, каждый из фильмов имеет свою величину c_i , характеризующую, насколько он интересен. В кинотеатре сегодня n последовательных сеансов фильмов. Вы можете пойти на несколько подряд идущих сеансов. Если вы посмотрите фильм ровно один раз, то получите от него удовольствие. Если вы посмотрите фильм больше чем один раз или не посмотрите вообще ни разу, то удовольствие вы не получите. Общее удовольствие это сумма c_i фильмов, которые вы посмотрели ровно один раз.

Какое максимальное общее удовольствие вы можете получить?

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq m \leq n \leq 10^6$). Вторая строка содержит n целых чисел t_i ($1 \leq t_i \leq m$) — сеансы фильмов, в том порядке, в котором они демонстрируются в кинотеатре. Третья строка содержит m целых чисел c_i ($1 \leq c_i \leq 10^6$) — интерес фильмов.

Формат выходных данных

Выведите одно число — ответ на задачу.

Пример

стандартный ввод	стандартный вывод
8 4 1 3 2 2 4 2 1 4 3 5 6 6	15

Замечание

В примере вы можете посмотреть сеансы со второго по седьмой, тогда фильмы 1, 3 и 4 вы посмотрите ровно по одному разу.

Задача I. Слишком разноцветная матрица

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

Вам дана матрица $n \times m$, состоящая из целых чисел от 1 до 256 — описание картинки, где каждый элемент это цвет пикселя.

Вам не нравится, что в матрице слишком много разных цветов. Вы можете выбрать некоторый прямоугольник площадью не больше s пикселей и выключить все пиксели, покрытые им. Определите, какое наименьшее количество различных цветов после этого может остаться в матрице.

Формат входных данных

Первая строка содержит три целых числа n, m и s ($1 \leq n, m \leq 1000, 1 \leq s \leq nm$). Следующие n строк содержат по m целых чисел a_{ij} ($1 \leq a_{ij} \leq 256$) — цвета пикселей.

Формат выходных данных

Выведите одно число — ответ на задачу. Если можно покрыть матрицу целиком, выведите 0.

Примеры

стандартный ввод	стандартный вывод
1 11 7 1 5 4 1 4 4 1 3 3 1 3	2
3 3 4 1 2 3 1 3 4 1 1 1	1

Замечание

В первом примере, если закрыть пиксели со второго по восьмой, то будут видны пиксели только с цветами 1 и 3.

Задача J. Опасная лента

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

Перед вами опасная лента, которая разделена на n отрезков единичной длины. С одной стороны лента целиком покрыта химикатом, с другой стороны первые a отрезков и последние b отрезков покрыты химикатом. Ленту можно согнуть по точке (сгиба) **между** отрезками ровно на 180 градусов, однако если в процессе сгиба (или нескольких сгибов) химикат соприкоснется с химикатом (не важно, на какую сторону он был нанесен), то лента взорвется.

Вы хотите определить, сколькими способами можно согнуть ленту так, чтобы она не взорвалась. Два способа согнуть считаются различными, если существует точка сгиба, в которой лента согнута в одном способе, и не согнута или согнута в другом направлении в другом.

Формат входных данных

Даны три целых числа n, a, b ($a, b > 0, a + b \leq n \leq 1000$).

Формат выходных данных

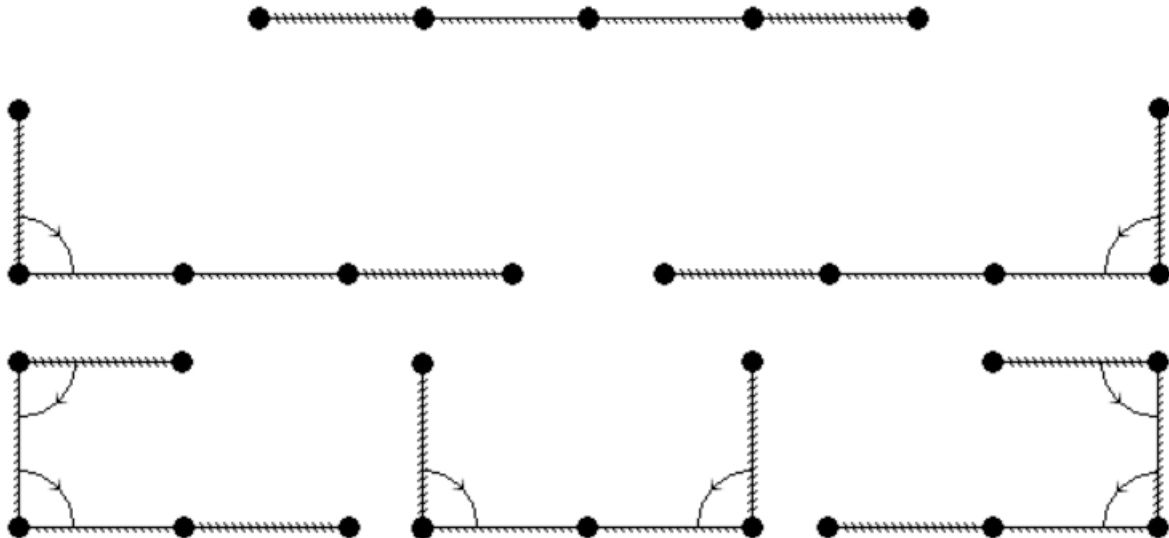
Выведите одно число — ответ на задачу по модулю $10^9 + 7$.

Примеры

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

Замечание

В первом примере существует шесть способов согнуть ленту безопасно, они показаны на картинке. Для того чтобы лучше понять процесс, на картинке повороты только на 90 градусов, в реальности они поворачивают на 180 градусов.



Задача К. Изучаем мьютексы

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Одним из интереснейших предметов на Физтехе считается курс по параллельным и распределённым вычислениям. Тихон очень любит разрабатывать многопоточные бэкэнд-сервисы. В таком сервисе несколькими потоками иногда может потребоваться читать и записывать одни и те же структуры данных в памяти. Чтобы гарантировать, что все потоки имеют корректное представление одной структуры данных, можно использовать так называемые *мьютексы* для защиты доступа к ней.

Мьютекс — это объект, который потоки могут захватывать и освобождать. Когда мьютекс захвачен, он не может быть захвачен снова, пока не будет освобождён — даже тем же потоком! Если поток попытается захватить мьютекс, который уже захвачен им, поток заблокируется, ожидая, пока он сам освободит мьютекс.

Тихон написал программу, состоящую из ряда функций. Каждая функция состоит из списка команд, которые выполняются последовательно при вызове функции. Каждая из команд — это:

- захватить мьютекс по имени X
- освободить мьютекс по имени X
- получить доступ к структуре данных, защищённой мьютексом X
- вызвать другую функцию

Предполагая, что функция `main` вызывается в начале программы, вы должны проверить, что:

1. всякий раз, когда требуется доступ к структуре данных, защищённой мьютексом X , мьютекс X в настоящее время захвачен;
2. всякий раз, когда требуется получить мьютекс, программа ещё не захватила его (чтобы избежать блокировки);
3. всякий раз, когда требуется освободить мьютекс, программа в настоящее время удерживает его.

Формат входных данных

Первая строка содержит число n — количество функций. Дальше идут описания каждой из n функций.

Описание функции начинается с целого числа m ($m \geq 1$) и строки s , что означает, что есть функция с именем s , которая содержит m команд. За этим следует m строк, каждая из которых содержит команду. Команды могут иметь следующий вид:

- `acquire X` — захватить мьютекс с именем X
- `release X` — освободить мьютекс с именем X
- `access X` — осуществляется доступ к структуре данных, которая защищена мьютексом X
- `call F` — вызывается функция с именем F

Все функции и мьютексы имеют имена длиной от 1 до 10 символов, содержащие только символы `a-z`. Ни одна из двух функций не будет иметь одинаковые имена, и всегда будет функция с именем `main`. Гарантируется, что бесконечной рекурсии не будет: функция никогда не вызывает себя ни напрямую, ни через цепочку других функций.

Суммарное число команд по всем функциям не превышает 50 000.

Формат выходных данных

Если программа не содержит ошибок, выведите `ok`.

В противном случае выведите первую ошибку, которая возникает во время выполнения. А именно:

1. если к структуре данных осуществляется доступ без захвата правильного мьютекса, выведите `corruption`,
2. если происходит блокировка, выведите `deadlock`,
3. если мьютекс освобождается, не будучи захваченным, выведите `error`.

Примеры

стандартный ввод	стандартный вывод
3 5 main acquire m call split acquire m call split acquire m 2 split access m release m 1 merge release m	ok
3 2 a acquire x call b 2 b access x acquire y 2 main call a call b	deadlock
1 2 main release x acquire x	error
1 1 main access x	corruption

Задача L. Пересекающиеся отрезки

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Перед вами $2n$ точек на плоскости. Разобьем их на n пар и соединим точки в паре отрезком. Требуется, чтобы все пары получившихся отрезков пересекались.

Формат входных данных

В первой строке дано целое число T ($1 \leq T \leq 100$) — количество тестовых случаев. Каждый тестовый случай начинается со строки, содержащей целое число n ($2 \leq n \leq 10^5$). Следующие $2n$ строк содержат целые числа, по модулю не превосходящие 10^9 — координаты точки. Никакие три точки не лежат на одной прямой. Гарантируется, что $\sum n$ по всем тестовым случаям не превосходит 10^5 , а также что можно разбить точки на пары требуемым образом.

Формат выходных данных

Для каждого тестового случая выведите одну перестановку p_i длины $2n$, которая обозначает, что точка i соединена с точкой p_i (обратите внимание, что если $p_i = j$, то $p_j = i$).

Пример

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

Задача М. Рейтинг стартовой точки

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

Перед вами матрица размера $n \times m$, в каждой клетке матрицы записано целое число a_{ij} . Некоторые из этих клеток помечены как стартовые. Рейтинг стартовой точки — это минимальное число d , такое что вы можете достичь как минимум t клеток матрицы, начав в соответствующей стартовой точке. Перемещаться по матрице можно только в клетки, соседние по стороне, и только если разница чисел в соответствующих клетках не превосходит d .

Некоторые клетки отмечены как стартовые. Для каждой из таких клеток вы должны найти её рейтинг.

Формат входных данных

Первая строка содержит три целых числа n, m, t ($1 \leq n, m \leq 500, 0 \leq t \leq nm$). i -я из следующих n строк содержит по m целых чисел a_{ij} ($0 \leq a_{ij} \leq 10^9$) — числа, записанные в матрице. Следующие n строк содержат по m целых чисел — единицу, если клетка является стартовой, и ноль, если нет.

Формат выходных данных

Для каждой стартовой клетки выведите её рейтинг. Необходимо выводить ответы для клеток в порядке увеличения i , при равенстве i в порядке увеличения j .

Пример

стандартный ввод	стандартный вывод
3 3 5 1 6 0 11 15 30 6 23 59 1 0 0 0 0 1 0 0 0	9 15

Задача N. Сортировка матрицы

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

Вам дана матрица $n \times m$, с числами a_{ij} . Каждое число от 1 до nm записано в матрице ровно один раз.

Вы совершаете три операции с матрицей:

1. Для каждой строки матрицы переупорядочите элементы в строке так, как вы этого хотите.
2. Для каждого столбца матрицы переупорядочите элементы в столбце так, как вы этого хотите.
3. Для каждой строки матрицы переупорядочите элементы в строке так, как вы этого хотите.

С помощью этих трех операций вы хотите сделать так, чтобы выполнялось равенство $a_{ij} = m(i - 1) + j$. Покажите способ достичь этого.

Формат входных данных

В первой строке даны целые числа n и m ($1 \leq n, m \leq 100$). Каждая из следующих n строк содержит по m целых чисел a_{ij} ($1 \leq a_{ij} \leq nm$). Гарантируется, что есть способ привести матрицу к требуемому виду.

Формат выходных данных

Выведите сначала матрицу, которая получится после первой операции, а затем матрицу, которая получится после второй операции. Каждая матрица должна содержать n строк по m чисел.

Примеры

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