Задача A. Расстояние Левенштейна

Имя входного файла:	`distance.in`
Имя выходного файла:	`distance.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Дана текстовая строка. С ней можно выполнять следующие операции:

- Заменить один символ строки на другой символ.
- Удалить один произвольный символ.
- Вставить произвольный символ в произвольное место строки.

Например, при помощи первой операции из строки «СОК» можно получить строку «СУК», при помощи второй операции — строку «ОК», при помощи третьей операции — строку «СТОК». Минимальное количество таких операций, при помощи которых можно из одной строки получить другую, называется стоимостью редактирования или расстоянием Левенштейна.

Определите расстояние Левенштейна для двух данных строк.

Формат входных данных

Программа получает на вход две строки, длина каждой из которых не превосходит 5000 символов, строки состоят только из заглавных латинских букв.

Формат выходных данных

Требуется вывести одно число — расстояние Левенштейна для данных строк.

Примеры

distance.in	distance.out
ABCDEFGH ACDEXGIH	3

Задача B. Робот в серверной

Имя входного файла:	`servers.in`
Имя выходного файла:	`servers.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вы — системный администратор одной очень перспективной IT-фирмы. В чём именно заключаются перспективы этой фирмы никто не знает, но все уверены, что они есть. Под вашим управлением находится серверная, в которой установлено целых n серверов (или n целых серверов, вы точно не уверены), пронумерованных целыми числами от 1 до n.

Ваша работа — следить за тем, чтобы серверы работали и не выключались. Если какой-нибудь сервер выключится, вы должны моментально его включить (по крайней мере, так считает ваш начальник). Вашу работу усложняет тот факт, что работоспособность некоторых серверов зависит от других серверов (например, для работы Web-сервера должен работать сервер баз данных). Если попытаться включить сервер, когда работают не все сервера, от которых он зависит, то этот сервер просто не включится (и потом придется пытаться включать его заново). Например, если сервер номер 1 зависит от серверов $2, 3$ и 5, то для того, чтобы его включить, требуется сначала включить серверы $2, 3$ и 5.

Как истинный системный администратор, вы ленивы и поэтому решили переложить эту сложную задачу на плечи маленького ни в чём не повинного робота. Бедняга робот не смог вам отказать, поэтому каждый день он выполняет следующий алгоритм. У робота в памяти записана последовательность из n различных чисел от 1 до n — порядок, в котором он обходит серверы. Подходя к очередному серверу, робот проверяет, выключен ли сервер, и если он выключен, то пытается его включить. После того, как робот дошёл до последнего сервера, он заканчивает выполнение алгоритма.

Недавно в серверной отключилось электричество, поэтому сейчас все серверы выключены. Вам интересно — сколько раз роботу придется выполнить свой алгоритм, чтобы все серверы оказались включенными.

Формат входных данных

Первая строка ввода содержит число n — количество серверов в серверной ($1 \leqslant n \leqslant 10^5$). Следующая строка содержит n различных чисел от 1 до n — порядок, в котором робот обходит серверы. В следующих n строках задано описание зависимостей между серверами: i–я строка содержит число k_i — количество серверов, от которых зависит i–й сервер, и последовательность k_i различных чисел — номера этих серверов. Сервер не может зависеть от самого себя. Суммарное количество зависимостей не превышает $1\,000\,000$.

Формат выходных данных

В единственную строку выведите количество итераций алгоритма, через которое робот включит все серверы, или -1, если робот так и не сможет включить все серверы.

Примеры

servers.in	servers.out
3 3 1 2 1 2 0 0	2
3 2 1 3 1 2 1 3 1 1	-1

Замечание

Подробное описание действий в первом примере:

1. Робот подходит к серверу 3 и включает его, так как этот сервер не зависит от других серверов;
2. Робот подходит к серверу 1 и не включает его, так как ещё не включен сервер 2, от которого зависит сервер 1;
3. Робот подходит к серверу 2 и включает его, так как этот сервер не зависит от других серверов;
4. Заканчивается первая итерация;
5. Робот подходит к серверу 3, сервер уже включен;
6. Робот подходит к серверу 1 и включает его, так как все серверы, от которых он зависит, уже включены;
7. Робот подходит к серверу 2, сервер уже включен;
8. Все серверы включены, прошло две итерации;

Во втором примере невозможно включить все серверы.

Задача C. Защищенное соединение

Имя входного файла:	`secure.in`
Имя выходного файла:	`secure.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В свете недавних новостей о прослушке каналов связи, два непримиримых интернет-гиганта Урагании «Laim.UR» и «Xenda» решили подписать соглашение об установлении защищенного канала связи между дата-центрами друг друга. В Урагании n городов, но, к сожалению, ни в одном городе нет дата-центров обоих гигантов. Поэтому для формирования защищенного канала придется прокладывать междугородние линии связи.

Специалисты компаний определили m пар городов, которые можно соединить, проложив сегмент канала связи, и оценили стоимость создания такого сегмента для каждой из этих пар.

Результирующий канал может состоять из нескольких сегментов. Он должен начинаться в одном из городов, где находится дата-центр первой компании, может проходить через промежуточные города и должен заканчиваться в городе, где находится дата-центр второй компании.

Теперь необходимо определить минимальную стоимость защищенного канала, соединяющего два дата-центра компаний.

Формат входных данных

В первой строке находятся целые числа n и m ($2 \leqslant n \leqslant 5\,000$, $1 \leqslant m \leqslant 10^5$) — количество городов и количество пар городов, которые можно соединить сегментом канала связи.

Во второй строке находятся n целых чисел a_i ($0 \leqslant a_i \leqslant 2$). Если $a_i = 0$, то в i-м городе нет дата-центра ни одного из гигантов. Если $a_i = 1$, то в i-м городе есть дата-центр «Laim.UR», а если $a_i = 2$, то в i-м городе находится дата-центр «Xenda». Гарантируется, что среди этих чисел есть как минимум одна единица и одна двойка.

В каждой из следующих m строк находится по три целых числа — s_i, t_i и c_i, которые означают, что города s_i и t_i ($1 \leqslant s_i, t_i \leqslant n$, $s_i \neq t_i$) можно соединить сегментом канала связи стоимостью c_i ($1 \leqslant c_i \leqslant 10^5$). Каждую пару городов можно соединить не более чем одним сегментом канала.

Формат выходных данных

Если соединить защищенным каналом связи два дата-центра разных интернет-гигантов возможно, то выведите в выходной файл три числа: x, y и d, означающие, что между городами x и y возможно провести канал связи суммарной стоимостью d. В городе x должен находиться дата-центр «Laim.UR», в городе y — дата-центр «Xenda». Если существует несколько оптимальных ответов, выведите любой. Если провести искомый канал невозможно, выведите -1.

Примеры

secure.in	secure.out
6 7 1 0 1 2 2 0 1 3 3 1 2 4 2 3 3 2 4 2 1 6 5 3 5 6 5 6 1	3 4 5

Задача D. Гадание по-карельски Light

Имя входного файла:	`karelia-light.in`
Имя выходного файла:	`karelia-light.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Юкка очень любит гадать по звездам, но небо часто бывает облачным, поэтому для гадания используется карта звездного неба. На карте нарисовано N звезд. Для гадания необходимо выбрать некоторые K звезд, и если многоугольник с вершинами в этих точках является выпуклым, то гадание считается успешным. Многоугольник является выпуклым, если все его вершины будут находиться по одну сторону относительно прямой, проходящей через каждую из его сторон, или на ней.

Пекка, наблюдая за гаданиями Юкки, захотел узнать, сколько различных выпуклых K-угольников можно построить, выбирая их вершины из заданных точек.

Ваша задача — дать ответ на поставленный вопрос для нескольких возможных значений K.

Формат входных данных

В первой строке входного файла записаны два числа — количество точек N и количество рассматриваемых случаев L ($3 \leqslant N \leqslant 13$, $1 \leqslant L \leqslant N - 2$). В следующих N строках — по два целых числа X_i, Y_i — координаты точек ($-10\,000 \leqslant X_i, Y_i \leqslant 10\,000$). Никакие три точки не лежат на одной прямой и не совпадают.

Требуется подсчитать количество выпуклых многоугольников для L различных значений числа K, перечисленных в последней строке. Все L чисел — целые положительные, каждое из которых не менее трех и не более N.

Формат выходных данных

В первой строке выведите через пробел L чисел — ответы для каждого из случаев.

Примеры

karelia-light.in	karelia-light.out
5 3 0 0 4 4 0 4 4 0 2 3 3 4 5	10 3 0

Задача E. Катый ноль

Имя входного файла:	`kthzero.in`
Имя выходного файла:	`kthzero.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Реализуйте эффективную структуру данных, позволяющую изменять элементы массива и вычислять индекс k-го слева нуля на данном отрезке в массиве.

Формат входных данных

В первой строке вводится одно натуральное число N ($1 \leqslant N \leqslant 200\,000$) — количество чисел в массиве. Во второй строке вводятся N чисел от 0 до 100 000 — элементы массива. В третьей строке вводится одно натуральное число M ($1 \leqslant M \leqslant 200\,000$) — количество запросов. Каждая из следующих M строк представляет собой описание запроса. Сначала вводится одна буква, кодирующая вид запроса (`s` — вычислить индекс k-го нуля, `u` — обновить значение элемента). Следом за `s` вводится три числа — левый и правый концы отрезка и число k ($1 \leqslant k \leqslant N$). Следом за `u` вводятся два числа — номер элемента и его новое значение.

Формат выходных данных

Для каждого запроса s выведите результат. Все числа выводите в одну строку через пробел. Если нужного числа нулей на запрашиваемом отрезке нет, выводите -1 для данного запроса.

Примеры

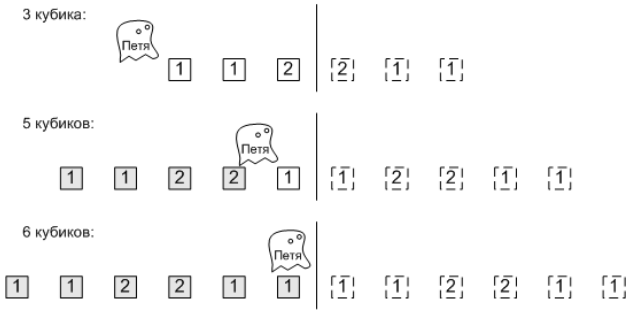
kthzero.in	kthzero.out
5 0 0 3 0 2 3 u 1 5 u 1 0 s 1 5 3	4

Задача F. Кубики

Имя входного файла:	`cubes.in`
Имя выходного файла:	`cubes.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Привидение Петя любит играть со своими кубиками. Он любит выкладывать их в ряд и разглядывать своё творение. Однако недавно друзья решили подшутить над Петей и поставили в его игровой комнате зеркало. Ведь всем известно, что привидения не отражаются в зеркале! А кубики отражаются.

Теперь Петя видит перед собой N цветных кубиков, но не знает, какие из этих кубиков настоящие, а какие — всего лишь отражение в зеркале.



Помогите Пете! Выясните, сколько у него может быть кубиков. Петя видит отражение всех кубиков в зеркале и часть кубиков, которая находится перед ним. Часть кубиков может быть позади Пети, их он не видит.

Формат входных данных

Первая строка входного файла содержит два целых числа: N ($1 \leqslant N \leqslant 100\,000$) и количество различных цветов, в которые могут быть раскрашены кубики,— M ($1 \leqslant M \leqslant 100\,000$). Следующая строка содержит N целых чисел от 1 до M — цвета кубиков.

Формат выходных данных

В выходной файл выведите в порядке возрастания все такие K, что у Пети может быть K кубиков.

Примеры

cubes.in	cubes.out
6 2 1 1 2 2 1 1	3 5 6