

Задача А. Фабрика

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

Машины, как известно, используют тепло и электрическую активность тел людей для получения энергии. У них даже имеются собственные фабрики для получения и обработки энергии.

На одной из такой фабрик энергия получается следующим образом. Люди находятся в двух комнатах: в первой комнате для получения энергии берется тепло человека, а во второй его электрическая активность. Для наиболее продуктивного процесса производства энергии требуется, чтобы в обеих комнатах было одинаковое количество людей.

К сожалению, час назад произошел бунт, после которого некоторые люди исчезли, а остальные перемешались. Теперь в первой комнате находится a человек, а во второй b . Машины хотят сделать так, чтобы в обеих комнатах было одинаковое число людей. Однако в силу своего программного обеспечения они могут только удваивать количество людей в комнате, где находится меньшее число людей за счет другой комнаты (то есть, если в первой комнате три человека, а во второй пять, то они за один шаг могут перегнать из второй комнаты в первую три человека). Могут ли они такими действиями уравнять количество людей в комнатах?

Формат входного файла

В первой строке дано число n ($1 \leq n \leq 10^5$) — количество тестов, на которые ваша программа должна дать ответ. Далее идут n строк по два числа a и b в каждой ($1 \leq a, b \leq 10^9$).

Формат выходного файла

Выведите n строк, в каждой из которых должно быть либо «YES», если машины смогут уравнять количество людей в соответствующем тесте, и «NO» если нет.

Примеры

factory.in	factory.out
3	YES
2 6	NO
1 5	YES
4 4	

Задача В. Огромные прыжки

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

Команда «Навуходоносора» во главе с Морфеусом обучает Нео совершать огромные прыжки. Морфеус разработал учебный план, выполнив который Нео будет готов к битве с агентами. План заключается в следующем.

Всего Нео необходимо сделать n прыжков в том порядке, в котором ему будет удобнее это сделать. Длина i -го прыжка составляет a_i метров. Перед каждый прыжком Нео необходимо потратить несколько (возможно, 0) часов на тренировки. Нео знает, что для выполнения i -го прыжка без тренировок ему необходимо вполнить перед i -ым прыжком хотя бы t_i других прыжков. Если же Нео этого не сделает, ему придется дополнительно тренироваться a_i часов перед тем, как выполнить i -ый прыжок из плана.

Нео решил, что выполнять прыжки он будет в том порядке, при котором он затратит как можно меньшее количество часов на тренировки перед прыжками. Помогите ему — посчитайте минимальное возможное число часов, которое Нео придется потратить на тренировки.

Формат входного файла

В первой строке входного файла дается одно целое число n ($1 \leq n \leq 10^5$) — количество прыжков, которые необходимо выполнить Нео. Следующие n строк содержат по два целых числа a_i и t_i ($0 \leq a_i \leq 10^9$, $0 \leq t_i \leq n$) — длина прыжка и количество прыжков, необходимых для выполнения прыжка без подготовки соответственно.

Формат выходного файла

В первую строку выходного файла выведите одно целое число — минимальное количество часов, которое Нео придется уделить тренировкам.

Примеры

<code>jumps.in</code>	<code>jumps.out</code>
3	2
5 2	
2 2	
1 1	

Комментарий

В приведенном примере Нео необходимо выполнить второй прыжок, потратив два часа на тренировки. После этого он сможет, не тренируясь, выполнить третий прыжок, а после него — и первый.

Задача С. Каждая пони умеет писать

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

Две недели назад принцесса Селестия попросила всех пони научиться писать к званому вечеру во дворце Селестии. Эпплджек все эти две недели так была занята на своей ферме, что совсем забыла об этом. Осталась всего пара часов, а она понятия не имеет как написать хоть какое-нибудь слово. Хорошо, что Твайлайт Спаркл оказалась неподалеку.

«Вот кто сможет выручить меня»,— подумала Эпплджек и обратилась к Твайлайт за помощью. Началось обучение.

В течение часа Эпплджек пыталась вникнуть в это занимательное мастерство, но, к сожалению, смогла овладеть лишь написанием одного слова.

«Как же мне быть, сахарок? Принцесса Селестия прознает о том, что я умею писать всего одно слово»,— с расстройством сказала Эпплджек.

«На самом деле ты умеешь писать не одно слово. Все слова в нашем языке обладают одним интересным свойством: если s — слово, а p — префикс s , то sp — также является словом. Но обычно пони не пишут слишком длинные или короткие слова, слова длины n — считаются наиболее вежливыми и пони стараются писать только их»,— ответила Твайлайт.

«И много различных слов длины n я смогу написать?» — спросила Эпплджек.

«Ну это надо посчитать. Ты полностью поняла алгоритм построения слов?» — спросила Твайлайт и задумалась.

«Да, конечно, это же совсем просто. Кстати, что такое префикс?» — радостно ответила Эпплджек.

Пока Твайлайт объясняет Эпплджек, что такое префикс, посчитайте количество различных слов, которые Эпплджек сможет написать. Без вашей помощи они не успеют.

Формат входного файла

Первая строка входного файла содержит целое число n ($1 \leq n \leq 5000$) — длины слов, которые необходимо получить. Вторая строка содержит строку длины k ($1 \leq k \leq n$), состоящую из строчных латинских букв — единственная строка, которую знает Эпплджек.

Формат выходного файла

В выходной файл выведите единственно число — количество различных строк, которые сможет получить Эпплджек из исходной. Разумеется, после того, как узнает, что такое префикс. Количество нужно вывести по модулю $10^9 + 7$. (Что такое модуль Эпплджек, конечно же, знает).

Примеры

<code>prefixstrings.in</code>	<code>prefixstrings.out</code>
12 abacaba	10

⁴ Задача D. Революция

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

Решающая битва Нео и Смита назначена на 19:00, но из-за сбоя системы Смит-Оракул не знает точное время, и поэтому агент Смит опаздывает. Нео в это время решил изучить оставшихся Смитов, стоящих в ряд.

Известно, что Агент Смит, превращая кого-либо в себя, не меняет код полностью. Поэтому, изучая исходники любого Смита, можно сказать, был человек перед превращением мужчиной или женщиной. Нео вспомнил, как тяжело на корабле, когда мужчин гораздо больше, чем женщин, и решил вычислить количество подотрезков из агентов Смитов, которые, по мнению Нео, являются k -хорошими.

Нео считает, что отрезок A является k -хорошим, если он начинается со Смита-женщины, кончается Смитом-мужчиной и содержит внутри $(k - 1)$ -хороший подотрезок, ни одна из границ которого не совпадает с границей отрезка A . При этом 1-хорошим называется любой отрезок, который начинается с женщины и заканчивается мужчиной.

Формат входного файла

В самой первой строке написано число m — число тестов. В каждом тесте в первой строке заданы числа n и k ($2 \leq n \leq 100000$, $1 \leq k \leq n/2$), где n — количество агентов Смитов в ряду. Во второй строке заданы n не разделенных пробелами чисел a_i ($0 \leq a_i \leq 1$), где $a_i = 0$, если Смит был женщиной, и $a_i = 1$, если Смит был мужчиной.

Суммарное количество агентов во входном файле не превышает 100000.

Формат выходного файла

Для каждого теста выведите количество подотрезков, которые являются k -хорошими, но не являются $k + 1$ -хорошими.

Пример

<code>revolutions.in</code>	<code>revolutions.out</code>
2	0
3 1	2
110	
3 1	
011	

Задача Е. Стражи

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

Морфеуса в Матрице схватили агенты, а права на вождение антигравитационными кораблями есть только у него. Поэтому Тринити не справилась с управлением и завела «Навуходоносор» в тупик, который является прямоугольником из $n \times m$ секторов. В каждом секторе, за исключением того, куда попал корабль, находится Страж.

Стражи сделаны в компании «Matrixsoft», поэтому Страж не видит корабль, если зионское расстояние от стража до корабля больше k . То есть, если страж находится в секторе с координатами (a, b) , а корабль — в секторе с координатами (x, y) , то страж не видит корабль, если $|a-x|+|b-y| > k$, и увидит, если $|a-x|+|b-y| \leq k$. Страж нападет на корабль только в том случае, если он его видит.

Посчитайте количество Стражей, которые нападут на корабль.

Формат входного файла

Первая строка входного файла содержит два целых числа n и m ($1 \leq n, m \leq 1000$) — размеры отсека, где застрял «Навуходоносор». Во второй строчке находятся два числа x и y ($1 \leq x \leq n, 1 \leq y \leq m$) — координаты сектора, в котором находится корабль. В третьей строчке находится число k ($1 \leq k \leq 2000$) — расстояние, на котором Стражи видят корабль.

Формат выходного файла

Выведите одно целое число: количество стражей, которые нападут на «Навуходоносор».

Примеры

<code>sentinels.in</code>	<code>sentinels.out</code>
2 2 1 1 1	2

Задача F. Смит

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

Как известно, агент Смит — обычная программа, и, чтобы ее взломать, не нужно делать чего-то экстраординарного, нужно всего лишь реализовать алгоритм Андерсона. Сам алгоритм очень сложен в написании, и хоть сколько-нибудь быстрая его работа возможна лишь на серверах Матрицы. Однако, он основывается на куда более простом алгоритме Томаса, который мы и попросим вас реализовать.

Нам даны n чисел, отвечающих за жизнеобеспечение агента. Поскольку некоторые органы могут быть повреждены, числа могут быть отрицательными. Чтобы вызвать каскадный резонанс и уничтожить агента, нужно выбрать несколько чисел таким образом, чтобы их произведение было больше произведения всех остальных.

Нео написал алгоритм, решающий эту задачу, примерно за 10 секунд. Повторите его подвиг.

Формат входного файла

В первой строке находится число n ($2 \leq n \leq 1000$) — количество чисел. Во второй строке находятся n целых чисел a_i ($-100 \leq a_i \leq 100$) — числа отвечающие за жизнеобеспечение агента Смита.

Формат выходного файла

В первой строке выведите количество чисел, которые необходимо выбрать. Во второй строке выведите индексы этих чисел.

Примеры

<code>smith.in</code>	<code>smith.out</code>
5	1
5 3 -1 15 2	1

Задача G. Матрица

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

База данных агентов Смитов состоит из n строк, каждая из которых состоит из k столбцов. В данной задаче мы считаем, что каждое поле в таблице это целое число. Эта база была очень хорошо структурирована и отлично выполняла свои задачи. Ровно до того момента, пока злобный хакер Тринити не проникла в нее и испортила её стройную структуру. То, что она сделала, было воистину кощунственно: она пересортировала таблицу по-своему!

Для этого она зафиксировала некоторую перестановку p чисел от 1 до k и действовала следующим образом: упорядочила строки по возрастанию столбца с номером p_1 , при равных значениях в столбце p_1 , по возрастанию чисел в столбце p_2 , и так далее. Для восстановления порядка агентам необходимо знать перестановку p . Помогите агентам Смитам!

Формат входного файла

В первой строке даны числа n и k ($1 \leq n, k \leq 10^3$) — количество строк и столбцов в таблице. В каждой из следующих n строк содержится по k целых чисел a_{ij} — элементы таблицы. ($1 \leq a_{ij} \leq 2 \times 10^3$)

Формат выходного файла

Выведите k различных чисел от 1 до k — искомую перестановку. В случае нескольких возможных перестановок выведите любую. Если такой перестановки не существует, выведите -1.

Примеры

<code>table.in</code>	<code>table.out</code>
3 3 2 3 1 3 2 1 1 2 2	3 1 2
<code>table.in</code>	<code>table.out</code>
2 2 2 1 1 1	-1

Задача Н. Башни

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

Нео и Морфеус очень любят играть в одну настольную игру. Правила этой игры довольно сложные и мы не будем их описывать в условии в полной мере. На карте для этой игры в два ряда расположены башни. Некоторые башни верхнего ряда соединены дорогами с башнями нижнего ряда. В начале каждый игрок владеет ровно одной башней. Причем первый игрок владеет башней из верхнего ряда, а второй — из нижнего. Цель каждого игрока — захватить все башни.

У каждой башни есть своя цена. Для каждой игры стоимость башен выбирается случайным образом, чтобы не получалось так, что всегда выигрывает, например, первый игрок. Нео и Морфеус заметили, что игра получается совсем неинтересной, если в самом начале игры башни, которые им принадлежат, соединены дорогой. Также они заметили, что чем больше суммарная стоимость башен, которыми они владеют на первом ходу, тем игра получается интересней.

Морфеусу и Нео надоело самим выбирать начальное положение для игры, и они попросили у вас помощи. Пронумеруем башни верхнего ряда слева направо начиная с единицы. Аналогично сделаем и с башнями нижнего ряда. Каждая дорога соединяет две различные башни напрямую. Известно, что никакие две дороги не пересекаются. Более формально это можно записать следующим образом: для дороги, которая соединяет башню i верхнего ряда и башню j нижнего, нельзя найти дорогу, которая соединяет башни k и l , такие, что $i > k$ и $j < l$, или такие, что $i < k$ и $j > l$. Вам необходимо найти такую пару башен, что сумма их стоимостей максимальна и они не соединены дорогой.

Формат входного файла

В первой строке дано три числа n, m, k ($1 \leq n, m \leq 10^5, 0 \leq k \leq 10^5$) — количество башен в верхнем ряду, количество башен в нижнем ряду и количество дорог. В следующих k строках находится по два числа a и b ($1 \leq a \leq n, 1 \leq b \leq m$), которые обозначают, что существует дорога между башней a верхнего ряда и башней b нижнего. В следующей строке записано число t ($1 \leq t \leq 100$) — количество различных наборов стоимостей башен. Каждый набор характеризуется четверкой чисел a, b, c, d ($1 \leq a, b, c, d \leq 10^6$). Она обозначает следующее. Стоимость первой башни первого ряда равна a . Пусть стоимость i -й башни равна c_i , тогда стоимость $i + 1$ равна $(c_i \cdot b + c) \% d$. При генерации стоимостей башен будем считать, что после последней башни верхнего ряда следует первая башня нижнего ряда.

Формат выходного файла

Для каждого набора стоимостей необходимо вывести в отдельной строке два числа — номера башен из верхнего и нижнего ряда, которые удовлетворяют условию. Если же таких башен не существует, выведите в этой строке -1. Если существует несколько ответов, которые удовлетворяют условию, можете вывести любой.

Примеры

<code>towers.in</code>	<code>towers.out</code>
1 1 1	-1
1 1	-1
2	
1 2 3 4	
5 6 7 8	

towers.in	towers.out
4 5 3	1 1
1 2	2 5
1 3	1 5
4 5	1 1
4	
1 2 3 4	
10 20 30 40	
10 9 8 7	
7 2 4 199	

Задача I. Навуходоносор

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Это интерактивная задача.

Неизвестные хакеры пытаются взломать систему управления кораблём «Навуходоносор». Они уже получили доступ к панели управления оружием. Всего на вооружении «Навуходоносора» имеется n ЭМИ-пушек. Для управления пушкой оператор садится в специальное кресло, номер которого совпадает с номером пушки, и управляет пушкой, сидя в этом кресле.

Навуходоносор неплохо защищён от взлома, и пока что хакеры научились делать только одно действие. Заключается оно в том, что хакеры могут выбрать любые два кресла, которые управляют некоторыми пушками, после чего поменять местами пушки, которыми эти хакеры управляют. Так, если они выбирают кресла i и j , управляющие пушками с номерами a_i и a_j соответственно, то после работы хакеров кресло номер i будет управлять пушкой номер a_j , а кресло номер j — пушкой с номером a_i .

Вашей программе необходимо возглавить работу команды корабля по нейтрализации хакеров. В перерывах между двумя действиями хакеров команда успевает сделать по два аналогичных действия. Программа должна сообщать команде, какие две пары кресел им необходимо переключить, после чего она будет получать информацию о том, какую пару кресел переключают хакеры. В тот момент, когда правильные настройки будут восстановлены (каждое кресло будет управлять пушкой с таким же номером), программе будет необходимо сообщить об этом команде и завершить работу.

Протокол взаимодействия с программой жюри:

В первой строке программа жюри сообщает вашей программе одно целое число n ($1 \leq n \leq 100$) — количество пушек на «Навуходоносоре». В следующей строке следуют n различных чисел a_i ($1 \leq a_i \leq n$) — пушки, которые управляются соответствующими креслами в данный момент. Далее до момента завершения работы программы повторяются следующие действия.

Ваша программа выдает четыре натуральных числа, не превышающих n , по два на строке — две пары кресел, которые необходимо переключить команде. После этого программа жюри передает в отдельной строке два натуральных числа, не превышающих n — номера кресел, которые переключила команда хакеров.

Если перед каким-то из ваших действий все кресла правильно подключены к пушкам, вместо очередного действия необходимо вывести `Done` и завершить выполнение программы. Суммарно вами должно быть сделано не более 5000 действий.

Пример

stdin	stdout
3	2 3
3 2 1	1 3
1 2	Done

Для корректной работы программы после каждой операции вывода данных вам необходимо делать следующие операции:

- В языке Pascal: `flush(output);`
- В C/C++: `fflush(stdout);`
- В Java: `System.out.flush();`
- В Python: `sys.stdout.flush();`

Задача J. Таблетки от Морфеуса

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайта

Это интерактивная задача.

Известный всем способ определения избранных, примененный Морфеусом в случае с Нео, безнадежно устарел. Теперь вместо двух таблеток разного цвета кандидатам в избранные предлагают 12 таблеток, цвет которых одинаков.

Чтобы несколько упростить эту проверку, одна из этих таблеток отличается по весу от всех остальных. То, в какую сторону она отличается, неизвестно. Кандидату в избранные дается возможность провести три взвешивания на весах без гирь. После этого кандидат должен назвать таблетку, отличающуюся по весу от всех остальных, и спасти мир. Напишите программу, проходящую этот тест вместо кандидата.

Протокол взаимодействия с программой жюри:

Ваша программа должна вывести слово `start` в отдельной строке. После этого трижды повторяются следующие действия.

Ваша программа выводит в отдельной строке таблетки, которые хочет взвесить при очередном взвешивании в формате

$$\text{left: } l_1 l_2 \dots l_k \text{ right: } r_1 r_2 \dots r_t$$

где l_i — номера таблеток, которые необходимо положить на левую чашу весов, r_i — то же самое для правой чаши. Все числа, выведенные во время одного запроса, должны быть различны, и не должны превышать 12. После этого программа жюри выводит `less`, если левая чаша легче, чем правая, `more`, если правая чаша легче чем левая, и `equal`, если вес таблеток на обеих чашах одинаков.

После этого вашей программе необходимо вывести

$$\text{answer: } z$$

где z — номер искомой таблетки.

Примеры

stdin	stdout
<code>start</code>	<code>left: 1 2 3 4 right: 5 6 7 8</code>
<code>less</code>	<code>left: 1 2 right: 3 4</code>
<code>less</code>	<code>left: 1 right: 2</code>
<code>less</code>	<code>answer: 1</code>

Комментарий

Для корректной работы программы после каждой операции вывода данных вам необходимо делать следующие операции:

- В языке Pascal: `flush(output);`
- В C/C++: `fflush(stdout);`
- В Java: `System.out.flush();`
- В Python: `sys.stdout.flush();`