

Задача А. Покраска в два цвета

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Дан неориентированный граф из n вершин и m рёбер. Нужно покрасить вершины графа в два цвета таким образом, чтобы все рёбра соединяли вершины разных цветов, или выяснить, что это невозможно.

Формат входных данных

В первой строке ввода заданы через пробел два целых числа: число вершин n и число рёбер m ($1 \leq n \leq 100$, $1 \leq m \leq 5000$). Далее следует m строк, каждая из которых содержит по два целых числа в диапазоне от 1 до n — номера вершин, которые соединяет ребро.

Формат выходных данных

Если требуемой покраски не существует, выведите -1 . В противном случае выведите через пробел n целых чисел от 1 до 2 — цвета вершин. Если решений несколько, выведите любое из них.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 3 1 2 2 3 3 1	-1
3 1 1 2	1 2 2

Задача В. Двудольный граф

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

В этой задаче вам нужно построить двудольный граф, который обладает следующими свойствами:

1. Количество вершин в первой доле равно d .
2. Количество вершин во второй доле равно $d - 2$.
3. Количество рёбер не должно превышать $3d$.
4. Для любого двудольного графа, полученного удалением двух вершин из первой доли, должно существовать совершенное паросочетание.

Напомним, что двудольным называется граф, в котором вершины разделены на две доли так, что каждое ребро соединяет вершину первой доли с вершиной второй доли. Совершенное паросочетание — это такой набор рёбер, что каждая вершина графа является концом ровно одного ребра из этого набора.

Формат входных данных

В первой строке задано целое число T — количество тестовых случаев ($1 \leq T \leq 100$). Каждая из следующих T строк содержит одно целое число d — количество вершин в первой доле в данном тестовом случае ($3 \leq d \leq 100$).

Формат выходных данных

Для каждого тестового случая сначала на отдельной строке выведите число m — количество рёбер. В следующих m строках выведите описание рёбер, по одному на строке. Каждое ребро описывается парой чисел u и v — номерами вершин первой и второй доли, соединённых этим ребром ($0 \leq u < d$, $0 \leq v < d - 2$).

В графе не должно быть кратных рёбер.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1	7
4	0 1
	1 0
	1 1
	2 0
	2 1
	3 0
	3 1

Замечание

Количество рёбер минимизировать не требуется.

Задача С. Друзья и враги

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

На острове Дженту живут n крокодилов. Каждый крокодил имеет ровно одного друга и ровно одного врага среди остальных крокодилов на острове. Отношения дружбы и вражды симметричны: если крокодил A — друг крокодила B , то крокодил B — друг крокодила A ; если же крокодил A — враг крокодила B , то и крокодил B — враг крокодила A . Никакой крокодил не является одновременно врагом и другом другого крокодила. Кроме того, никакой крокодил не может быть ни другом, ни врагом самому себе.

Крокодилы острова Дженту весьма эмоциональны. Когда встречаются два врага, они со страшной силой колотят по земле хвостами, а затем происходит Битва Крокодилов. Когда встречаются два друга, они исполняют разрушительный Танец Дружбы Крокодилов.

Недавно танцы и битвы разбудили вулкан, находящийся под островом, и остров раскололся на две части. Вулкан успокоился, но крокодилы опасаются, что дальнейшие танцы и битвы разбудят его снова, и обе половины острова зальёт лава.

Теперь крокодилы хотят расселиться по двум половинам острова таким образом, чтобы на каждой половине оказалось *нейтральное* множество крокодилов — такое множество, что никакие два крокодила в нём не являются ни друзьями, ни врагами. Расселившись так, крокодилы не будут устраивать танцы и битвы, а значит, можно надеяться, что вулкан не будет извергаться снова.

Крокодилы острова Дженту мудры, но непрактичны. Они понимают, что при таком устройстве дружбы и вражды разделение крокодилов на два нейтральных множества всегда возможно, но не знают, как именно разделиться на такие множества.

Помогите им! Найдите такое разбиение крокодилов на два множества, что у каждого крокодила в своём множестве нет ни друзей, ни врагов.

Формат входных данных

В первой строке задано натуральное число n — количество крокодилов ($4 \leq n \leq 100$). Следующие n строк описывают крокодилов. В первой из них записаны два числа f_1 и e_1 через пробел — номер друга первого крокодила и номер его врага. Во второй записаны числа f_2 и e_2 — номера друга и врага

второго крокодила, и так далее. В последней из этих строк записаны f_n и e_n — друг и враг крокодила с номером n . Крокодилы пронумерованы числами от 1 до n в том порядке, в котором они описываются во входных данных.

Все числа f_k и e_k целые и лежат в пределах от 1 до n , включительно. Отношения дружбы и вражды симметричны. Друг и враг каждого крокодила различны. Никакой крокодил не является ни другом, ни врагом самому себе.

Формат выходных данных

В первой строке выведите n чисел через пробел. Каждое из этих чисел должно быть равно либо 1, либо 2. Если i -е и j -е числа равны, это означает, что крокодилы с номерами i и j оказались на одной половине острова. Если же эти числа различны, значит, крокодилы i и j оказались на разных половинах острова.

Если правильных ответов несколько, можно вывести любой из них.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4	1 2 1 2
2 4	
1 3	
4 2	
3 1	

Задача D. Максимальное паросочетание

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Граф (V, E) называется *двудольным*, если множество его вершин V можно разбить на такие два подмножества A и B , что любое ребро из E соединяет вершину из A с вершиной из B .

Паросочетанием P называется любое подмножество E такое, что никакие два ребра из него не имеют общей вершины.

Максимальное паросочетание — это паросочетание, число рёбер в котором максимально.

Найдите максимальное паросочетание в заданном двудольном графе.

Формат входных данных

В первой строке заданы три числа m и n и k ($1 \leq m, n \leq 100$, $1 \leq k \leq 10\,000$), где m — число вершин в множестве A , n — число вершин в B , а k — количество рёбер в графе. Каждая из следующих k строк содержит по два числа u_i и v_i , означающих, что вершина u_i множества A соединена ребром с вершиной v_i множества B . Вершины во множествах A и B нумеруются по отдельности, начиная с единицы. Все числа во входных данных целые.

Формат выходных данных

В первой строке выведите число l — количество рёбер в максимальном паросочетании. Далее выведите l строк, по два числа в каждой. Числа a_j и b_j , стоящие в j -й из этих строк, означают, что в паросочетание взято ребро между вершиной a_j множества A и вершиной b_j множества B . Взятые рёбра должны образовывать максимальное паросочетание.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 3 4	2
1 1	1 1
1 2	2 3
2 2	
2 3	

Задача Е. Волшебные шляпы

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Орден Всяческих Острых и Магических Штукоев заказал изготовление нескольких волшебных шляп. Волшебная шляпа получается, если соединить два конуса: декоративный верхний конус и пушистый тёплый нижний конус. Чтобы сделать шляпу, сначала оба конуса кладут на стол таким образом, чтобы их основания были горизонтальны, а вершины указывали точно вверх. Затем верхний конус поднимают и опускают сверху на нижний конус. При этом основание верхнего конуса должно оставаться горизонтальным.

Не всякая пара конусов годится для того, чтобы изготовить волшебную шляпу. Пара конусов, из которых получается настоящая волшебная шляпа, должна удовлетворять двум критериям:

- Вершина верхнего конуса должна быть строго выше вершины нижнего конуса. Другими словами, если верхний конус поместить над нижним так, чтобы оси конусов совпадали, а затем опустить вдоль общей оси, их вершины не будут касаться.
- У шляпы должны получиться поля, состоящие из части нижнего конуса. Другими словами, какая-то часть нижнего конуса, имеющая ненулевую площадь, должна быть видна из-под верхнего конуса. В противном случае шляпа будет выглядеть в точности как верхний конус, а не как настоящая волшебная шляпа!

У вас есть несколько верхних конусов и несколько нижних конусов различных размеров. Каждый конус характеризуется высотой (расстоянием от основания до вершины по оси) и радиусом основания. Определите, какое максимальное количество волшебных шляп можно изготовить из имеющихся в наличии конусов.

Формат входных данных

Первая строка ввода содержит целое число m — количество верхних конусов ($1 \leq m \leq 50$). Следующие m строк содержат описания верхних конусов. Каждая из этих строк содержит два целых числа h_i^u и r_i^u — высоту и радиус соответствующего верхнего конуса ($1 \leq h_i^u, r_i^u \leq 10\,000$). В следующей строке находится целое число n — количество нижних конусов ($1 \leq n \leq 50$). Далее

идут n строк, описывающих нижние конусы. Каждая из них содержит два целых числа h_j^d и r_j^d — высоту и радиус соответствующего нижнего конуса ($1 \leq h_j^d, r_j^d \leq 10\,000$).

Формат выходных данных

Выведите одно число: максимальное количество волшебных шляп, которые можно изготовить.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1 30 3 1 3 30	1
2 4 4 4 3 2 5 5 12 4	1
1 3 3 2 1 2 1 4	1
2 10 2 10 5 2 2 3 9 6	2

Задача F. Минимальное контролирующее множество

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Требуется построить в двудольном графе минимальное контролирующее множество, если дано максимальное паросочетание.

Формат входных данных

В первой строке даны два числа m и n — размеры долей ($1 \leq m, n \leq 4000$). Каждая из следующих m строк содержит список рёбер, выходящих из соответствующей вершины первой доли. Этот список начинается с числа K_i ($0 \leq K_i \leq n$) — количества рёбер, после которого записаны вершины второй доли, соединённые с данной вершиной первой доли, в произвольном порядке. Сумма всех K_i во входных данных не превосходит 500 000. Последняя строка содержит некоторое максимальное паросочетание в этом графе — m чисел L_i ($0 \leq L_i \leq n$) — соответствующая i -й вершине первой доли вершина второй доли, или 0, если i -я вершина первой доли не входит в паросочетание.

Формат выходных данных

Первая строка содержит размер минимального контролирующего множества. Вторая строка содержит количество вершин первой доли S , после которого записаны S чисел — номера вершин первой доли, входящих в контролирующее множество, в возрастающем порядке. Третья строка содержит описание вершин второй доли в аналогичном формате.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 2	2
2 1 2	1 1
1 2	1 2
1 2	
1 2 0	

Задача G. Доминошки

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Задана прямоугольная доска, некоторые клетки из которой вырезаны. Определите, можно ли покрыть оставшиеся клетки доминошками.

Формат входных данных

В первой строке заданы два целых числа m и n ($1 \leq m, n \leq 40$) — размеры доски. Каждая из следующих m строк содержит по n символов: i -й символ j -й из этих строк равен «X» (большая буква икс), если клетка вырезана, и «.» (точка), если она пуста.

Формат выходных данных

Выведите «YES», если доску можно покрыть доминошками, и «NO» в противном случае.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 3 ... XXX	NO
3 2 .X .. X.	YES

Задача Н. Любимые сыновья

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Давным-давно в одной стране жил-был король, и было у него n сыновей. Король хотел женить своих сыновей на девушках, которых они любили. Поэтому однажды он позвал сыновей в свои покои и попросил рассказать ему, в кого они влюблены.

Но все сыновья были ещё слишком юны и не могли принять окончательное решение о том, кого именно они любят. Вместо этого они сказали королю, каких девушек они считают красивыми, но, поскольку сыновья были разные, их выбор тоже не совсем совпал.

Король был мудр. Он записал, кто из сыновей каких девушек считает красивыми, после чего призвал вас — своего главного волшебника.

— Я хотел бы, чтобы все мои сыновья были счастливы, — сказал король, — но, поскольку это может оказаться невозможно, я хочу, чтобы хотя бы кто-нибудь из них женился на девушке, которую он считает красивой. Установите, кому на ком жениться.

Внезапно вы вспомнили, что не так давно король рассказывал вам про каждого из своих сыновей. Из этих рассказов вы знаете, насколько велика любовь короля к каждому из них. Имея это в виду, вы решили организовать женитьбы так, чтобы король был как можно более счастлив. Счастье короля пропорционально квадратному корню из суммы квадратов величины любви короля к каждому из сыновей, который женится на девушке, которую он считает красивой.

Решите, кто на ком должен жениться, так, чтобы максимизировать счастье короля.

Формат входных данных

В первой строке ввода содержится целое число n — количество сыновей короля ($1 \leq n \leq 400$). Во второй строке заданы n целых чисел a_i через пробел — величины любви короля к его сыновьям ($1 \leq a_i \leq 1000$).

Следующие n строк содержат списки девушек, которых сыновья короля считают красивыми. В i -й из этих строк содержится список, предоставленный i -м сыном. Каждый список начинается с целого числа k_i — количества девушек в нём, после которого следуют через пробел k_i различных целых чисел — номера девушек. Все потенциально красивые девушки королевства

(их в те времена было немного) пронумерованы целыми числами от 1 до n .

Формат выходных данных

Выведите через пробел n чисел — для каждого сына номер девушки, которая ему нравится и на которой он должен жениться, или 0, если ему придётся жениться на девушке, которой нет в его списке красивых.

Если обозначить как L множество тех сыновей, которые женятся на девушках из своих списков, ваш ответ должен максимизировать величину

$$\sqrt{\sum_{i \in L} a_i^2}.$$

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4	2 1 0 4
1 3 2 4	
4 1 2 3 4	
2 1 4	
2 1 4	
2 1 4	

Задача I. Каждая девушка хочет на бал

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Дни становятся всё длиннее, ночи всё короче, а экзамены всё ближе и ближе. Но совершенно не надвигающиеся государственные экзамены волнуют 11-классниц третьей педагогической гимназии города Нуоченьдалековска. Сразу после зимних каникул всем было объявлено, что в этом году выпускной вечер будет проходить в форме бала, как и мечтали все девушки, начиная с первого класса.

Конечно же, каждая девушка очень тщательно готовится к этому событию: шьёт себе платье, изучает танцевальные движения, присматривается к юношам — ведь разве можно пойти на бал без партнёра?

Примерно к концу апреля кто-то вдруг сообразил, что, помимо того, что среди выпускников N юношей и M девушек, и числа эти далеко не обязательно будут равны к концу года, многие девушки согласятся пойти на бал далеко не с каждым юношей, как, впрочем, и наоборот.

Более точно, каждому участнику бала соответствует число, определяющее уровень его привлекательности, установленный общественным мнением и число, определяющее его требовательность. Юноша может пойти с девушкой на бал, только если его привлекательность не ниже её требовательности, и наоборот.

Взволнованные подобными догадками выпускницы попросили директора школы (умудрённого опытом и, безусловно, авторитетного в подобных вопросах) разобраться в сложившейся непростой ситуации. Поскольку для директора все ученики равны, то его интересует вопрос, какое максимальное количество пар можно составить. И помните, что каждая принцесса достойна попасть на бал!

Формат входных данных

В первой строке входных данных содержится число N — количество юношей в выпускных классах. Следующие N строк содержат описания юношей в формате $b_i w_i$ — уровень привлекательности и требовательность, соответственно. В следующей строке содержится число M — количество девушек. Следующие M строк содержат описания девушек в том же формате, что и описания юношей.

Ограничения: $1 \leq N, M \leq 300\,000$, $1 \leq b_i, w_i \leq 10^9$.

Формат выходных данных

В первой строке выведите величину ответа K — максимальное количество пар участников бала, которые можно составить. Разумеется, помимо величины ответа, директор школы хочет знать и сам ответ, то есть кто с кем должен встать в пару. Поэтому в следующих K строках выведите составленные пары в любом порядке. Первое число пары — номер юноши во входных данных. Аналогично, второе число пары — номер девушки во входных данных.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2	2
1 3	1 2
3 1	2 1
2	
1 3	
3 1	
4	1
1 1	1 2
3 3	
5 5	
7 7	
2	
4 4	
1 1	

Задача J. Король

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

И было у мужика три дочери...

В Тридесятом царстве, Тридевятом государстве жил-был король. И было у короля n дочерей. В Тридесятом царстве жили n прекрасных юношей, и король знал, какие юноши нравятся каждой дочери (поскольку дочери были молодыми и бесшабашными, то им могли нравиться несколько юношей одновременно).

Однажды король приказал своему советнику подобрать для каждой дочери прекрасного юношу, за которого та сможет выйти замуж. Советник выполнил приказ и подобрал для каждой дочери для замужества прекрасного юношу, который ей нравился. Разумеется, каждый юноша может жениться только на одной из дочерей.

Посмотрев на список женихов, король сказал: «Мне нравится этот список, но я хочу знать для каждой дочери список всех юношей, за которых она может выйти замуж. Разумеется, при этом все остальные дочери также должны сохранить возможность выйти замуж за юношей, которые им нравятся».

Эта задача оказалась для советника слишком сложной. Помогите ему избежать казни, решив её.

Формат входных данных

Первая строка содержит число n — количество дочерей ($1 \leq n \leq 2000$). Следующие n строк содержат списки прекрасных юношей, которые нравятся дочерям. В начале идёт k_i — количество юношей, которые нравятся i -й дочери. Затем идут k_i чисел — номера юношей. Сумма k_i не превышает 200 000.

Последняя строка входных данных содержит список, составленный советником — n различных чисел от 1 до n : для каждой дочери — номер прекрасного юноши, за которого она может выйти замуж. Гарантируется, что список корректен — то есть каждой девушке нравится выбранный для неё юноша.

Формат выходных данных

Выведите n строк. Для каждой девушки выведите l_i — количество различных юношей, за которых она может выйти замуж. После этого выведите l_i чисел — номера этих юношей в произвольном порядке.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4	2 1 2
2 1 2	2 1 2
2 1 2	1 3
2 2 3	1 4
2 3 4	
1 2 3 4	