

Задача А. Задача о назначениях

Имя входного файла: `assignment.in`
Имя выходного файла: `assignment.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 Мб

Одной из классических задач комбинаторной оптимизации является так называемая задача о назначениях. Формулируется она следующим образом.

Есть n работников, пронумерованных числами от 1 до n , и n работ, также пронумерованных числами от 1 до n . Если i -ый работник выполняет j -ую работу, то ему выплачивается зарплата в размере c_{ij} денежных единиц. Необходимо найти такое назначение работников на работы (каждый работник выполняет ровно одну работу, каждая работа выполняется ровно одним работником), что суммарная зарплата работников минимальна (соответствующая сумма называется стоимостью назначения).

Напишите программу, решающую задачу о назначениях.

Формат входного файла

Первая строка входного файла содержит целое число n ($1 \leq n \leq 150$). Последующие n строк содержат по n чисел каждая. При этом j -ое число ($i+1$)-ой строки равно c_{ij} ($1 \leq c_{ij} \leq 1000$).

Формат выходного файла

В выходной файл выведите ответ на задачу — минимальную возможную стоимость назначения.

Пример

<code>assignment.in</code>	<code>assignment.out</code>
2 1 2 2 1	2
2 1 2 3 4	5

Задача В. Золушка

Имя входного файла: `cinderella.in`
Имя выходного файла: `cinderella.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

В некотором царстве, в некотором государстве жила Золушка. И была у бедной девушки злая мачеха, которая ее ужасно притесняла. Каждый день Золушке приходилось выполнять работу по дому. Уезжая на бал, мачеха приказала Золушке перебрать N видов крупы, находящихся в N тарелках, так чтобы крупинки одного вида лежали в отдельной тарелке. Золушке предстоит перебрать до 50 видов крупы. Хотелось бы определить какое минимальное количество крупинок M следует переложить, чтобы выполнить всю работу.

Ваша задача — написать программу, которая посчитает искомое число M .

Формат входного файла

В первой строке находится число N ($1 \leq N \leq 50$) — количество видов крупы. Последующие N строк содержат по N чисел, которые показывают, сколько крупинок каждого вида в i -той тарелке.

Формат выходного файла

Выведите в выходной файл одно число — искомое количество крупинок.

Пример

<code>cinderella.in</code>	<code>cinderella.out</code>
3 1 2 3 4 5 6 7 8 9	30

Задача С. Ограниченный поток

Имя входного файла: `flow.in`
Имя выходного файла: `flow.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Пусть задана некоторая транспортная сеть, необходимо вычислить максимальный поток, удовлетворяющий ограничению, в соответствии с которым поток через каждую вершину не должен превышать некоторые фиксированные пропускные способности.

Ваша задача — определить максимальный поток в заданной сети (графе, определенном выше) из первой вершины в последнюю с учетом ограничений на пропускные способности для вершин.

Формат входного файла

В первой строке входного файла указано одно целое число n , ($1 \leq n \leq 100$) — число вершин графа. Следующая строка содержит n чисел — ограничения на пропускную способность каждой вершины. Далее идет матрица смежности, в которой ненулевые элементы означают дугу из вершины i в вершину j с пропускной способностью, равной значению элемента матрицы.

Формат выходного файла

Выведите величину максимального потока в сети.

Пример

<code>flow.in</code>	<code>flow.out</code>
4 100 5 100 100 0 10 3 0 0 0 3 5 0 0 0 7 0 0 0 0	8

Задача D. Обобщенная задача о назначениях

Имя входного файла: `generalised.in`
Имя выходного файла: `generalised.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Известная задача о назначениях формулируется следующим образом: задана матрица размера $n \times n$. Необходимо выбрать по одному элементу в каждой строке так, чтобы в каждом столбце было выбрано ровно по одному элементу, и сумма выбранных элементов была бы минимальной.

Давайте обобщим задачу. Рассмотрим матрицу размера $n \times n$. Выберем k элементов в каждой строке так, чтобы в каждом столбце было выбрано ровно k элементов, и сумма выбранных элементов была бы минимально возможной.

Формат входного файла

В первой строке входного файла содержатся n и k ($1 \leq k \leq n \leq 50$). Следующие n строк содержат по n целых чисел — строки матрицы. Элементы матрицы не превосходят 10^3 по абсолютной величине.

Формат выходного файла

Выведите одно число — значение минимально возможной суммы элементов, выбранных описанным выше способом.

Пример

generalised.in	generalised.out
4 2 1 2 3 4 2 3 4 5 1 3 5 7 5 7 2 4	22

Формат входного файла

В первой строке входного файла два числа N и M ($1 \leq N, M \leq 100$). Вторая строка содержит количества детей в городах. В третьей строке задается число мест в лагерях. Далее следуют N строк по M чисел — стоимости путевок.

Формат выходного файла

В выходной файл выведите одно число — стоимость всех распределенных путевок.

Пример

rest.in	rest.out
3 3 40 23 20 20 20 43 10 5 4 6 4 5 7 3 6	355

Задача Е. Максимальный поток

Имя входного файла: `maxflow.in`
Имя выходного файла: `maxflow.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Одной из задач теории графов является задача определения максимального потока, протекающего от некоторой вершины s графа к некоторой вершине t . Вершина s называется истоком, а вершина t — стоком. При этом каждой дуге $\langle i, j \rangle$ (граф ориентированный) приписана некоторая пропускная способность $C(i, j)$, определяющая максимальное значение потока, который может протекать по данной дуге.

Ваша задача — определить максимальный поток в заданной сети (графе, определенном выше) из первой вершины в последнюю.

Формат входного файла

В первой строке входного файла указаны два целых числа n , ($1 \leq n \leq 200$) — число вершин графа и m — число дуг. Затем на m строках перечислены дуги графа. Каждая строка содержит три числа: начало дуги, конец дуги и пропускную способность дуги, которая не превышает 1000.

Формат выходного файла

Выведите величину максимального потока в сети

Пример

maxflow.in	maxflow.out
4 5 1 2 10 1 3 3 2 3 3 2 4 5 3 4 7	11

Задача F. Детский отдых

Имя входного файла: `rest.in`
Имя выходного файла: `rest.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Ежегодно сотни детей со всех населенных пунктов Республики отправляются отдыхать в M летних лагерей. В каждом из N городов проживает определенное количество ребят. Естественно, лагеря не могут принять всех желающих, потому что у любого из них свой предел на количество мест. Стоимость летнего отдыха ребенка зависит от конкретного лагеря плюс дорожные расходы. Большинство родителей не в состоянии оплачивать отдых своих детей, нет денег. Правительство Республики решило сделать ребятам подарок, и, сократив расходы на чиновников, изыскало средства для отправки школьников в лагерь.

Руководители проекта, занимаясь распределением путевок, должны определить такую схему распределения ребят по лагерям и бюджетных средств, чтобы за летний период отдохнуло как можно больше детей за меньшие деньги.

Задача G. Война

Имя входного файла: `war.in`
Имя выходного файла: `war.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Между двумя государствами А и В разразилась война. Обе стороны осуществляют материальное снабжение войск через систему взаимосвязанных дорог: автомобильных и железных.

Противник может прервать снабжение войск, нанося по дорогам бомбовые удары.

Руководство ашников поручило своим ученым определить на основе имеющихся разведданных о транспортной системе бэшников, какое минимальное число бомб нужно сбросить, чтобы лишить войска противника снабжения?

Формат входного файла

В первой строке входного файла содержится одно положительное число N — количество узлов в транспортной сети противника ($1 \leq N \leq 100$). Далее в N строках находится матрица смежности транспортной системы. Каждая единица матрицы — наличие дороги между пунктами i и j .

Формат выходного файла

В выходной файл выведите одно число — минимальное число бомб, необходимых для того, чтобы хоть один транспортный узел оказался отрезан от остальных.

Пример

war.in	war.out
3 0 1 1 1 0 1 1 1 0	2
3 0 1 0 1 0 0 0 0 0	0

Задача H. Обобщенный морской бой

Имя входного файла: `battle.in`
Имя выходного файла: `battle.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

В игре «обобщенный морской бой» корабль на доске $m \times n$ представляет собой вертикально или горизонтально ориентированный прямоугольник, ширина которого — одна клетка, а длина произвольна. Кораблей может быть

произвольное количество, они могут пересекаться или примыкать один к другому произвольным образом.

Располагая определенными разведанными относительно положения кораблей, попробуйте определить, каким может быть их наименьшее число.

Формат входного файла

Первая строка исходных данных задачи — размерность матрицы $1 \leq m, n \leq 100$, последующие m строк состоят из n нулей и единиц, единица соответствует занятой кораблем клетке, нули — клеткам, про которые неизвестно, принадлежат они какому-либо кораблю или нет.

Формат выходного файла

Результат вычислений — искомое наименьшее количество кораблей.

Пример

battle.in	battle.out
3 5 0 0 0 1 0 0 0 0 1 0 1 1 0 1 1	2

Задача I. Загрузка оборудования

Имя входного файла: equip.in
Имя выходного файла: equip.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Вася проходит производственную практику в вычислительной лаборатории университета, где обслуживает сложнейшую мультипроцессорную систему, параллельно решающую m задач для нужд университета и m — для нужд организаций-подрядчиков. Работы выполняет суперкомпьютер, $m \times m$ процессоров которого составляют матрицу из m строк и m столбцов. Процессоры специализированы, поэтому их производительность неодинакова. Она выражается целым числом a_{ij} миллионов операций в секунду, выполняемых процессором (i, j) .

Для решения каждой из «внутриуниверситетских» задач с номером $i \in 1..m$ требуются все процессоры с номерами $(i, j) \quad j \in 1..m$, причем все они в равной степени загружены этой задачей (т. е. на каждом из этих процессоров этой задачей выполняется одинаковое число операций в секунду), для решения внешней задачи $j \in 1..m$ задействуются процессоры $(i, j) \quad i \in 1..m$ тем же образом.

Помогите Васе распределить процессорное время между задачами так, чтобы суммарная производительность системы по всем задачам была наибольшей.

Формат входного файла

Первая строка исходного файла содержит число $1 \leq m \leq 100$ — размерность матрицы суперкомпьютера, последующие m строк — производительности процессоров.

Формат выходного файла

Первая строка вывода содержит искомую суммарную загрузку системы задачами. Вторая строка содержит m чисел, разделенных пробелами — i -ое число есть число миллионов операций, выполняемых i -ой университетской задачей на каждом из процессоров (i, j) (напомним, что для различных j эти числа совпадают). Третья строка содержит аналогичную информацию о задачах организаций-подрядчиков. Все числа следует выводить с 6 знаками после запятой.

Пример

equip.in	equip.out
3 4 5 6 5 7 9 6 9 12	19.000 3.000 5.000 6.000 0.000 2.000 3.000

Задача J. Проблема падишаха

Имя входного файла: pairs.in
Имя выходного файла: pairs.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Мудрый падишах внимательно следит за благополучием своих подданных, когда вершит их судьбы. В частности, на нем все заботы о вступающих в брачный возраст юношах и девушках его страны. И, как положено серьезному правителю, все по науке — перед тем, как творить молодые семьи, падишах провел Глобальное тестирование и по 100-балльной шкале определил совместимость всех юношей и девушек в совместном браке.

А дальше что? Падишах наслышан про задачу о назначении, но ему не нравится ее установка. Действительно, может ли быть спокойна его душа даже в случае всеобщего благополучия, если кому-то из подданных плохо? И можно ли жертвовать интересами хотя бы одной семьи во благо общества? Конечно, нет!

Падишаху милее другая мысль. Он хочет создать семью таким образом, чтобы степень неблагополучия самой неблагополучной по итогам тестирования пары была наименьшей. А решить эту неклассическую задачу он просит наш клуб программистов. Помогите падишаху!

Формат входного файла

Первая строка данных задачи — два целых числа: $1 \leq m \leq 200$ — количество девушек и $1 \leq n \leq 200$ — количество юношей. Последующие m строк содержат по n целых чисел от 0 до 100 — коэффициент совместимости соответствующей пары (меньшее значение менее способствует супружеской жизни).

Формат выходного файла

Ответ — искомый наименьший балл, при котором возможно создание максимально возможного количества семейных пар.

Пример

pairs.in	pairs.out
3 4 90 30 78 20 70 55 81 72 60 42 87 12	72

Задача K. Минимальное контролирующее множество

Имя входного файла: set.in
Имя выходного файла: set.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Множество вершин V' графа $G(V, E)$, содержащееся в V , называется контролирующим, если каждое ребро графа инцидентно хотя бы одной вершине из V' . Тривиальным примером контролирующего множества является множество всех вершин V .

Ваша задача состоит в том, чтобы в данном двудольном графе найти какое-либо контролирующее множество вершин минимальной мощности.

Формат входного файла

В первой строке входного файла содержатся два числа ($1 \leq N, M \leq 200$, — количество вершин в первой и второй долях. Далее до конца файла в каждой строке содержатся номера смежных вершин из первой и второй доли.

Формат выходного файла

В выходной файл в первой строке выведите мощность минимального контролирующего множества. Во второй строке выведите через пробел количества вершин из первой и второй долей, входящих в контролирующее множество. В третьей строке через пробел выведите номера вершин первой доли, входящих в контролирующее множество, а в четвертой — номера вершин второй доли, входящих в это множество.

Пример

set.in	set.out
6 6 1 1 2 5 3 4 4 4 5 2 5 3 6 1 6 6	5 4 1 1 2 5 6 4
6 6 1 1 1 2 2 2 3 2 3 3 4 3 4 4 4 5 4 6 5 2 6 3	4 2 2 1 4 2 3

Задача L. Домино

Имя входного файла: domino.in
Имя выходного файла: domino.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 Мб

Из клеточного листа бумаги размера $N \times N$ вырезали некоторые клетки. Определите, какое максимальное количество доминошек (плиток размера 2×1) можно разместить на оставшихся клетках. Каждая доминошка занимает две соседние клетки. Ни одна доминошка не должна занимать клетку, на которой лежит другая доминошка.

Формат входного файла

В первой строке входного файла содержится целое число N — размер листа ($1 \leq N \leq 40$). Далее в N строках задается матрица из нулей и единиц. Каждая единица матрицы означает вырезанную клетку.

Формат выходного файла

В выходной файл выведите одно число — максимальное количество доминошек, которые можно разместить на листе.

Пример

domino.in	domino.out
2 1 1 1 0	0
3 0 0 0 0 0 0 0 0 0	4
4 0 0 0 0 1 0 1 1 0 0 0 1 1 0 1 1	3

Задача M. Кони

Имя входного файла: knights.in
Имя выходного файла: knights.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 Мб

Из шахматной доски размера $N \times N$ удалили некоторые клетки. Определите, какое максимальное количество шахматных коней можно разместить на оставшихся клетках так, чтобы они не били друг друга.

Формат входного файла

В первой строке входного файла содержатся два целых числа: N — размер листа ($1 \leq N \leq 200$) и M — количество вырезанных клеток ($0 \leq M \leq N^2$). Далее в M строках задаются координаты вырезанных клеток. Координаты верхней левой клетки $(1, 1)$, а правой нижней — (N, N) . Гарантируется, что вырезанные клетки не повторяются.

Формат выходного файла

В выходной файл выведите одно число — максимальное количество коней, которых можно разместить на доске.

Пример

knights.in	knights.out
3 2 1 1 3 3	5

Задача N. Минимальный разрез

Имя входного файла: mincut.in
Имя выходного файла: mincut.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

В теории графов сетью называется ориентированный граф, в котором каждая дуга (u, v) имеет неотрицательную пропускную способность C_{uv} , определяющую максимальное значение потока, который может протекать по данной дуге. Выделяются две вершины: источник s и сток t .

Разрезом в сети называется разбиение множества вершин на два подмножества таких, что исток s попадает в одно подмножество, а сток t — в другое.

Минимальным разрезом в сети называется разрез наименьшей пропускной способности.

Ваша задача — определить минимальный разрез в заданной сети, у которой истоком является первая вершина, а стоком — последняя.

Формат входного файла

В первой строке входного файла указано одно целое число n , ($1 \leq n \leq 200$) — число вершин графа. Далее идет матрица смежности, в которой ненулевые элементы означают дугу из

вершины i в вершину j с пропускной способностью, равной значению элемента матрицы, которое не превышает 1000.

Формат выходного файла

В первой строке выходного файла выведите величину пропускной способности минимального разреза сети. Во второй строке выведите количество дуг минимального разреза. В последующих строках выведите через пробел номера вершин, являющихся началом и концом соответствующих дуг разреза.

Пример

mincut.in	mincut.out
4	11
0 10 3 0	3
0 0 3 5	1 3
0 0 0 7	2 3
0 0 0 0	2 4

Задача О. Слоники

Имя входного файла: `bishops.in`
Имя выходного файла: `bishops.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 Мб

Вася очень любит играть в шахматы. Однажды его заинтересовало, какое максимальное количество слонов K можно поставить на шахматную доску, размером $N \times M$ так, чтобы они не атаковали друг друга. Но с этой задачей он быстро справился. И тогда он подумал, а что делать, если на доске уже стоят какие-то другие фигуры? Эта задача оказалась не такой легкой. Тогда он решил обратиться за помощью к своему другу Пете.

Стали они вместе думать, но так ничего и не придумали. Помогите ребятам решить эту задачу.

Формат входного файла

В первой строке входного файла заданы числа N и M ($1 \leq N, M \leq 50$). В следующих N строках записано по M символов, описывающих шахматную доску: ‘X’ означает занятую клетку, через которую слон ходить не может, а ‘.’ — свободную клетку.

Формат выходного файла

В выходной файл выведите единственное число K . (Слоны атакуют друг друга, если они стоят на одной диагонали, и между ними нет других фигур).

Пример

bishops.in	bishops.out
3 3	6
...	
.X.	
...	

Задача Р. Аренда автомобиля

Имя входного файла: `rentacar.in`
Имя выходного файла: `rentacar.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Билл является владельцем нескольких пунктов аренды автомобилей. Каждый желающий может взять машину напрокат в любом пункте и вернуть ее в любой пункт, необязательно начальный.

Чтобы убедиться, что в каждом пункте всегда достаточное количество автомобилей, в течение недели работники Билла перегоняют лишние автомобили из одного пункта в другой. Они всегда делают так, чтобы в начале недели в каждом

пункте было одинаковое количество машин. Поэтому каждую неделю Биллу приходится решать, какую машину отправить в какой пункт.

Для упрощения задачи предположим, что люди всегда возвращают взятый автомобиль в некоторый пункт в ту же неделю, в которую они взяли его напрокат.

Пусть для каждого пункта аренды известно, сколько автомобилей было взято напрокат и сколько вернули в этот пункт. Кроме того, известна информация о дорогах между пунктами. Найдите общий кратчайший путь, который должны проделать все машины, чтобы в начале недели в каждом пункте аренды было одинаковое количество автомобилей.

Формат входного файла

Первая строка входного файла содержит n и m — количество городов и количество дорог в стране соответственно ($1 \leq n \leq 100$, $0 \leq m \leq 5000$). У Билла имеется по одному пункту аренды в каждом городе.

Следующие n строк описывают пункты аренды автомобилей. Каждый пункт задан двумя числами: t_i and r_i — количество автомобилей, взятых и возвращенных в этот пункт в течение последней недели. Общее количество взятых автомобилей равно общему количеству возвращенных автомобилей и не превосходит 10^4 .

Следующие m строк описывают дороги. Все дороги двусторонние. Каждая дорога задана номерами городов, которые она соединяет, и своей длиной. Длины дорог — положительные целые числа, не превосходящие 10^3 . Ни одна дорога не соединяет какой-либо город с самим собой. Допускается наличие нескольких дорог между двумя городами.

Формат выходного файла

В выходной файл выведите одно число — общий кратчайший путь, который должны пройти все автомобили, чтобы количество машин в каждом пункте в начале недели было одинаковым. Если невозможно перегруппировать автомобили заданным образом, выведите число -1 .

Пример

rentacar.in	rentacar.out
3 3	47
1 10	
5 0	
5 1	
1 2 10	
2 3 4	
1 3 3	

Задача Q. Замкнутый маршрут

Имя входного файла: `roundtrip.in`
Имя выходного файла: `roundtrip.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Давным-давно жил да был мудрый Король, и он управлял Королевством. Жители Королевства были счастливы. В этом Королевстве было n городов, и некоторые из них были соединены дорогами. Всего в Королевстве m дорог, и ездить по ним можно в обе стороны.

Король был настолько мудр, что понимал, существование только одной столицы может быть опасным, так как враг внезапно может захватить ее. Поэтому кроме столицы был выбран еще один город, из которого Король мог управлять Королевством. Этот город называли секретной столицей.

Чтобы Король мог безопасно добраться из столицы в секретную столицу, дороги прокладывались таким образом,

что существовало не менее трех различных путей, которые Король мог использовать. При этом никакой другой город не находился более, чем на одной из этих дорог.

Как-то раз Король решил объехать свое Королевство. Он собрался начать свое путешествие в столице и посетить два других города — секретную столицу и новый недавно построенный город. В конце поездки Король должен вернуться в столицу. По соображениям безопасности Король может посетить каждый город не более одного раза.

Помогите королевским советникам подготовить план поездки.

Формат входного файла

Первая строка входного файла содержит n и m ($4 \leq n \leq 10\,000$, $5 \leq m \leq 50\,000$). Города нумеруются от 1 до n . Вторая строка содержит три различных целых числа — a , b и c — номера столицы, секретной столицы и нового города соответственно.

Следующие m строк описывают дороги в Королевстве. Каждая дорога задана номерами городов, которые она соединяет. Между любыми двумя городами существует как минимум один путь. Никакой город не соединен дорогой с самим собой.

Формат выходного файла

Если не существует требуемого замкнутого маршрута, выведите -1 .

Иначе выведите любой маршрут. В первой строке выходного файла выведите k — количество дорог в маршруте. Во второй строке выведите $k + 1$ чисел — номера городов, посещенных Королем, в порядке посещения. Маршрут должен начинаться и заканчиваться в столице.

Пример

roundtrip.in	roundtrip.out
8 10	6
1 6 8	1 8 7 6 5 3 1
1 2	
1 3	
1 7	
1 8	
2 4	
3 5	
4 6	
5 6	
6 7	
7 8	

Задача R. Домино в казино

Имя входного файла: **casino.in**
Имя выходного файла: **casino.out**
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В домино очень любят играть во дворах после окончания рабочего дня. И это было так до тех пор, пока Джон Бигбак не представил игру в домино в своем казино “ПНДП” (Принесите нам деньги, пожалуйста).

Конечно, обычная игра в домино не очень подходит для казино, поэтому Джон решил изменить игру. Теперь игра проходит на прямоугольной доске размера $m \times m$. В каждой клетке доски записано некоторое целое число.

У игрока есть k костяшек домино — прямоугольников размера 2×1 . Игрок кладет их на доску так, чтобы они не пересекались с имеющимися. Сумма баллов подсчитывается как сумма произведений чисел под каждой костяшкой.

Например, есть два способа положить 2 костяшки на доску размера 2×2 . Для доски, приведенной ниже, лучший способ

показан слева. В этом случае сумма равна $1 \times 3 + 4 \times 2 = 11$. Если игрок положит костяшки так, как изображено справа, сумма будет равняться $1 \times 4 + 3 \times 2 = 10$.

1	4
3	2

Задана доска и количество костяшек, которые есть у игрока. Определите максимально возможную сумму, которую игрок может получить.

Формат входного файла

Первая строка входного файла содержит целые числа m , n и k ($1 \leq m \leq 16$, $1 \leq n \leq 100$, $1 \leq k \leq 200$). Следующие m строк содержат по n целых чисел, которые описывают доску. Числа на доске — положительные и не превышают 1000. Гарантируется, что на доске можно уложить все костяшки домино.

Формат выходного файла

Выведите одно целое число — максимально возможную сумму, которую игрок может получить.

Примеры

casino.in	casino.out
2 2 2	11
1 4	
3 2	

Задача S. Сумасшедшая стена

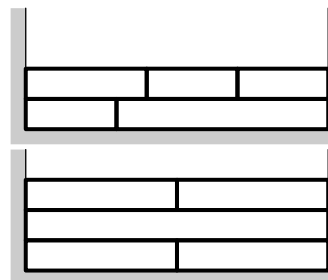
Имя входного файла: **wall.in**
Имя выходного файла: **wall.out**
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Авторы игры “Сумасшедший тетрис” решили создать новую игру, которую они назвали “Сумасшедшая стена”. Цель игры — построить стену, чтобы заблокировать проход. Стена должна быть максимально возможной высоты.

Ширина прохода w . Стена должна быть построена из нескольких кирпичей. Кирпичи падают сверху один за другим. Для каждого кирпича известны его левая и правая границы. Игрок не может перемещать кирпиче, но может выбрать порядок, в котором кирпичи падают. Кроме того, игрок может остановить игровой процесс и подтвердить, что игра окончена.

Стена состоит из нескольких рядов, ширина каждого из них w . Дополнительно, стена должна сплошной. Назовем стену сплошной, если никаких два кирпича не имеют одинаковые координаты левой границы, исключая ситуацию на левой стороне прохода, и никакие два кирпича не имеют одинаковые координаты правой границы, исключая ситуацию на правой стороне прохода.

Например, стена на рисунке слева сплошная, а справа — нет.



Задано описание имеющихся кирпичей, определите, сплошную стену какой максимально возможной высоты можно построить, а также какие кирпичи следует использовать.

Формат входного файла

Первая строка входного файла содержит n и m — количество кирпичей и ширину прохода соответственно ($1 \leq n \leq 2000$, $1 \leq w \leq 2 \cdot 10^9$). Следующие n строк описывают кирпичи. Каждый кирпич задает двумя целыми числами l_i и r_i — координата левой и правой границ ($0 \leq l_i < r_i \leq w$).

Формат выходного файла

В первой строке выходного файла выведите h — максимально возможную высоту стены. В следующих h строках выведите информацию о рядах кирпичей стены. Каждый ряд кирпичей должен быть описан номерами кирпичей, которые использовались для создания данного ряда. Перечисляйте кирпичи в порядке слева направо.

Примеры

wall.in	wall.out
6 10	2
0 3	1 4
3 7	5 6 3
7 10	
3 10	
0 4	
4 7	
5 10	2
0 5	1 4
0 5	3
0 10	
5 10	
5 10	

Задача Т. Газовая проблема

Имя входного файла: gas.in
Имя выходного файла: gas.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Соседняя страна Эджландия приняла решение перестать импортировать газ из Флатландии по причине очень высоких цен. После этого газовая компания Флатландии остановила подачу газа в газопровод, который проходит по территории Эджландии.

Газотранспортная система Эджландии состоит из газовых станций, соединенных трубами. Для каждой трубы известно, в каком направлении газ должен перекачиваться по трубе.

К сожалению, возникла некоторая проблема. Оказывается, полностью прекращать подачу газа не правильно. Если в трубе нет газа, то проверяющие и контролирующие системы могут быть повреждены.

Для каждой трубы известно минимальное количество газа, необходимого для того, чтобы поддерживать систему в работоспособном состоянии.

После длительных переговоров менеджеры газовой компании Эджландии приняли решение сохранить газотранспортную систему в рабочем состоянии. Технический газ должен циркулировать по трубам. Объем газа, поступающего на газовую станцию каждый день, должен равняться объему исходящего газа из этой станции. Итоговый объем газа, необходимого для осуществления проекта, равно сумме объемов газа, требуемых для каждой трубы каждый день.

Задана схема газотранспортной системы. Помогите газовой компании определить, какой минимальный объем газа требуется, чтобы система не вышла из строя.

Формат входного файла

Первая строка входного файла содержит n and m — количество газовых станций и газовых труб в газовой системе Эджландии ($2 \leq n \leq 300$, $2 \leq m \leq 1000$). Следующие m строк описывают газовые трубы. Каждая труба характеризуется тремя целыми числами — номерами газовых станций, которые труба соединяет и минимальным объемом газа, требуемого для нормальной работы системы. Минимальные объемы газа — целые числа, которые не превосходят 10^3 .

Для каждой станции существует не менее одной входящей трубы и не менее одной исходящей. Никакие две станции не соединены более, чем одной трубой. Если есть труба между двумя станциями, то не существует трубы, по которой газ идет в обратном направлении.

Формат выходного файла

В первой строке выходного файла выведите минимальный объем технического газа. Каждая следующая из m строк должна содержать объем газа, транспортируемого вдоль соответствующей трубы каждый день.

Если нет возможности сохранить газотранспортную систему в рабочем состоянии, выведите -1 в первой строке файла.

Примеры

gas.in	gas.out
4 5	10
1 2 1	1
2 3 1	1
1 3 1	2
4 1 3	3
3 4 3	3

Задача У. Две последовательности

Имя входного файла: seq.in
Имя выходного файла: seq.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Даны две числовые последовательности длины n и m . В каждой из последовательностей все числа попарно различны. За одну операцию можно удалить один произвольный элемент любой последовательности. Требуется определить минимальное число операций, за которое последовательности можно сделать одинаковыми.

Формат входного файла

Первая строка содержит два целых числа n , m ($0 \leq n, m \leq 100000$), разделенных пробелом. Во второй строке даны элементы первой последовательности, в третьей — элементы второй последовательности. Элементы каждой последовательности разделены пробелами. (Все элементы последовательностей — целые числа, не превосходящие по абсолютной величине 10^9).

Формат выходного файла

В выходной файл выведите минимальное число операций.

Пример

seq.in	seq.out
2 4	4
1 2	
-5 2 1 7	

Задача V. Спортивная площадка

Имя входного файла: `sport.in`
Имя выходного файла: `sport.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Кондопожскому ЦБК для строительства очередного грандиозного сооружения необходима ровная площадка прямоугольной формы. Но, поскольку рельеф местности холмистый, первый этап работы состоит в планировании участка, которое заключается в перемещении грунта и стоит весьма недешево.

Проделана немалая подготовительная работа. Вся территория площадки разбита на квадратные участки одинакового размера, для каждого из них задан объем грунта, избыточный — со знаком «+», недостающий — «-». После перемещения указанных объемов площадка станет совершенно ровной. А вот откуда и куда перемещать — отдел АСУ комбината не дает точных рекомендаций. Впрочем, на этапе подготовки сметы строительных работ вполне достаточно указать связанные с планировкой участка затраты.

Для решения этой нестандартной задачи решено привлечь студентов ПетрГУ, которые надеются на вашу помощь.

Формат входного файла

Первая строка входного файла содержит два целых числа — размеры $m \times n$ решетки, клетки которой составляют площадку ($1 \leq m, n \leq 60$). Далее следуют m строк по n разделенных пробелами целых чисел $-100 \leq a_{ij} \leq 100$ ($1 \leq i \leq m, 1 \leq j \leq n$), составляющих сведения об объемах перемещения грунта из клетки («+») или в клетку («-»).

Перемещение грунта осуществляется по вертикали или горизонтали в соседнюю клетку, затраты на такую операцию пропорциональны (условно — равны) объему перемещаемого грунта.

Формат выходного файла

В выходной файл выведите одно число — минимальные затраты на выполнение работ.

Пример

<code>sport.in</code>	<code>sport.out</code>
2 3 -50 6 -21 26 36 3	107