

Задача А. Кубики

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

Родители подарили Сене набор детских кубиков. Поскольку Сеня скоро пойдет в школу, они купили ему кубики с буквами. На каждой из шести граней каждого кубика написана буква.

Теперь Сеня хочет похвастаться перед старшей сестрой, что научился читать. Для этого он хочет сложить из кубиков ее имя. Но это оказалось довольно сложно сделать — ведь разные буквы могут находиться на одном и том же кубике и тогда Сеня не сможет использовать обе буквы в слове. Правда одна и та же буква может встречаться на разных кубиках. Помогите Сене!

Дан набор кубиков и имя сестры. Выясните, можно ли выложить ее имя с помощью этих кубиков и если да, то в каком порядке следует выложить кубики.

Формат входных данных

На первой строке входного файла находится число n ($1 \leq n \leq 100$) — количество кубиков в наборе у Сени. На второй строке записано имя Сениной сестры — слово, состоящее только из больших латинских букв, не длиннее 100 символов. Следующие n строк содержат по 6 букв (только большие латинские буквы), которые написаны на соответствующем кубике.

Формат выходных данных

На первой строке выходного файла выведите «YES», если выложить имя Сениной сестры данными кубиками можно, «NO» в противном случае.

Если ответ «YES», на второй строке выведите m различных чисел из диапазона от 1 до n , где m - количество букв в имени Сениной сестры, i -е число должно быть номером кубика, который следует положить на i -е место при составлении имени Сениной сестры. Кубики нумеруются с 1, в том порядке, в котором они заданы во входном файле. Если решений несколько, выведите любое. Разделяйте числа пробелами.

Примеры

<code>cubes.in</code>	<code>cubes.out</code>
4 ANN ANNNNN BCDEFG HIJKLM NOPQRS	NO
5 HELEN ABCDEF GHIJKL MNOPQL STUVWN EIUOZK	YES 2 5 3 1 4

Задача В. Испорченный паркет

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

Пол в некоторой комнате размером $M \times N$ замощен паркетом. При этом некоторые плитки паркета оказались испорчены. Петя решил сделать ремонт в этой комнате, заменив только испорченные клетки. Придя в магазин, он обнаружил, что паркетные плитки бывают двух типов — размера 1×2 , которые стоят A рублей (немного подумав, Петя понял, что плитки 1×2 можно поворачивать на 90 градусов, получая тем самым плитки 2×1) и размера 1×1 , которые стоят B рублей. Разрезать плитку размера 1×2 на две размера 1×1 Петя не может.

Определите, какая минимальная сумма денег нужна Пете, чтобы сделать ремонт.

Формат входных данных

Первая строка входного файла содержит 4 числа N, M, A, B ($1 \leq N, M \leq 300, A, B$ — целые числа, по модулю не превосходящие 1000). Каждая из последующих N строк содержит по M символов: символ «.» (точка) обозначает неиспорченную плитку паркета, а символ «*» (звездочка) — испорченную. В конце строк могут идти незначащие пробелы. В конце файла могут быть пустые строки.

Формат выходных данных

В выходной файл выведите одно число — минимальную сумму денег, имея которую можно заменить испорченные паркетины (и только их).

Примеры

floor.in	floor.out
2 3 3 2 .** .*.	5

Задача С. Минимальное контролирующее множество

Имя входного файла: `min-dominating-set.in`
Имя выходного файла: `min-dominating-set.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

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 входных данных

В первой строке входного файла содержатся два числа n и m — количество вершин в первой и второй долях ($1 \leq n, m \leq 200$). Далее до конца файла в каждой строке содержатся номера смежных вершин из первой и второй доли. В графе не бывает кратных рёбер.

Формат выходных данных

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

Примеры

<code>min-dominating-set.in</code>	<code>min-dominating-set.out</code>
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 D. День рождения

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

Митя знаком с m юношами и n девушками и хочет пригласить часть из них на свой день рождения. Ему известно, с какими девушками знаком каждый юноша, и с какими юношами знакома каждая девушка. Он хочет добиться того, чтобы каждый приглашённый был знаком со всеми приглашёнными противоположного пола, пригласив при этом максимально возможное число своих знакомых. Помогите ему это сделать!

Формат входных данных

Входной файл состоит из одного или нескольких наборов входных данных. В первой строке входного файла записано число наборов k ($1 \leq k \leq 20$). В последующих строках записаны сами наборы входных данных.

В первой строке каждого набора задаются числа $0 \leq m \leq 150$ и $0 \leq n \leq 150$. Далее следуют m строк, в каждой из которых записано одно или несколько чисел — номера девушек, с которыми знаком i -й юноша (каждый номер встречается не более одного раза). Строка завершается числом 0.

Формат выходных данных

Для каждого набора выведите четыре строки. В первой из них выведите максимальное число знакомых, которых сможет пригласить Митя. В следующей строке выведите количество юношей и количество девушек в максимальном наборе знакомых. Следующие две строки должны содержать номера приглашённых юношей и приглашённых девушек соответственно. Если максимальных наборов несколько, то выведите любой из них.

Примеры

<code>birthday.in</code>	<code>birthday.out</code>
2	4
2 2	2 2
1 2 0	1 2
1 2 0	1 2
3 2	4
1 2 0	2 2
2 0	1 3
1 2 0	1 2

Задача Е. Такси

Имя входного файла: `taxi.in`
Имя выходного файла: `taxi.out`
Ограничение по времени: 0.5 секунда
Ограничение по памяти: 256 мегабайт

Управлять службой такси — совсем не простое дело. Помимо естественной необходимости централизованного управления машинами для того, чтобы обслуживать заказы по мере их поступления и как можно быстрее, нужно также планировать поездки для обслуживания тех клиентов, которые сделали заказы заранее.

В вашем распоряжении находится список заказов такси на следующий день. Вам необходимо минимизировать число машин такси, необходимых чтобы выполнить все заказы.

Для простоты будем считать, что план города представляет собой квадратную решетку. Адрес в городе будем обозначать парой целых чисел: x -координатой и y -координатой. Время, необходимое для того, чтобы добраться из точки с адресом (a, b) в точку (c, d) , равно $|a - c| + |b - d|$ минут. Машина такси может выполнить очередной заказ, либо если это первый ее заказ за день, либо она успевает приехать в начальную точку из предыдущей конечной хотя бы за минуту до указанного срока. Обратите внимание, что выполнение некоторых заказов может окончиться после полуночи.

Формат входных данных

В первой строке входного файла записано число заказов M ($0 < M < 500$). Последующие M строк описывают сами заказы, по одному в строке. Про каждый заказ указано время отправления в формате `hh:mm` (в интервале с `00:00` по `23:59`), координаты (a, b) точки отправления и координаты (c, d) точки назначения. Все координаты во входном файле неотрицательные и не превосходят 200. Заказы записаны упорядоченными по времени отправления.

Формат выходных данных

В выходной файл выведите единственное целое число — минимальное количество машин такси, необходимых для обслуживания всех заказов.

Примеры

<code>taxi.in</code>	<code>taxi.out</code>
2 08:00 10 11 9 16 08:07 9 16 10 11	1
2 08:00 10 11 9 16 08:06 9 16 10 11	2

Задача F. Толстые хоббиты

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

Ни один хоббит не в состоянии в одиночку противостоять полчищам Мордора. . . В последний поход против Мордора Гэндальф решил отправить N хоббитов из Шира. Но часть хоббитов наотрез отказалась, жалуясь на то, что другие хоббиты наверняка будут дразнить их толстыми. После опроса всех хоббитов оказалось, что любой хоббит отказывается принять участие в походе в том случае, если с ним в поход выступит хотя бы один хоббит с меньшим весом. К счастью для Средиземья, не все хоббиты знают свой точный вес. В Шире были всего одни весы чашечного типа, позволяющие для пары хоббитов определить, какой хоббит тяжелее. Некоторые пары хоббитов взвешивались на этих весах. Всем хоббитам известен результат всех взвешиваний. Гэндальф абсолютно уверен, что в Шире нет двух хоббитов одного веса. Он заинтересован в том, чтобы отряд состоял из наибольшего количества хоббитов. Однако найти наибольшее множество хоббитов, среди которых ни один не считает себя тяжелее другого, оказалось не так-то просто. Подскажите Гэндальфу, на сколько хоббитов он может рассчитывать. Помните при этом, что хоббиты умные существа и знают, что если Сэм тяжелее Пиппина, а Пиппин тяжелее Фродо, то Сэм и подавно будет тяжелее Фродо.

Формат входных данных

В первой строке дано целое число N – количество хоббитов ($2 \leq N \leq 100$). Все хоббиты пронумерованы целыми числами от 1 до N . В следующих N строках записана матрица размера $N \times N$. Если i -й и j -й хоббит взвешивались на чашечных весах и оказалось, что i -й хоббит тяжелее, то в i -й строке матрицы на j -й позиции стоит единица. Во всех остальных случаях в матрице стоят нули.

Формат выходных данных

В первой строке выведите размер наибольшего множества хоббитов, готового выступить в поход, во второй строке перечислите номера хоббитов из этого множества через пробел.

Примеры

<code>hobbits.in</code>	<code>hobbits.out</code>
2 0 1 0 0	1 2
3 0 0 0 0 0 0 0 0 0	3 1 2 3

Задача G. Дракончики

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Петя занимается разведением дракончиков. У него есть M зеленых дракончиков и K желтых. У Пети есть N двухместных аквариумов для дракончиков и $M + K - 2N$ одноместных.

Петя, понаблюдав некоторое время за своими дракончиками, установил, что некоторые пары дракончиков не могут жить вместе (будучи помещенными в один аквариум они тут же начинают драться), а также некоторые дракончики совершенно не переносят одиночества и поэтому не могут жить в одноместном аквариуме.

Петя хочет с использованием своих знаний так разместить дракончиков по аквариумам, чтобы в каждом двухместном аквариуме обязательно был один зеленый дракончик и один желтый, и при этом драконы, не переносящие одиночества, обязательно были бы помещены в двухместный аквариум, и в двухместном аквариуме никогда не оказывалось бы двух драконов, которые не могут жить вместе.

Формат входных данных

Вводятся числа M, K, N ($M \geq 1, K \geq 1, N \geq 0, N \leq M, N \leq K, M + K \leq 200$). Будем считать, что зеленые дракончики пронумерованы числами от 1 до M , а желтые — числами от $M + 1$ до $M + K$.

Далее идет число T ($0 \leq T \leq M \cdot K$) — количество пар дракончиков, про которых известно, что они не переносят друг друга. Далее идет T пар чисел, описывающих номера не переносящих друг друга дракончиков (первое число каждой пары описывает зеленого дракончика, второе — желтого). Далее идет число Q ($0 \leq Q \leq M + K$) — количество дракончиков, не переносящих одиночества. Далее идет Q чисел, задающих номера этих драконов.

Формат выходных данных

В случае если разместить дракончиков по аквариумам требуемым образом нельзя, выведите единственное слово «NO». В противном случае первая строка должна содержать «YES». В следующие N строк выведите N пар чисел, задающих номера дракончиков, которых нужно поместить в двухместные аквариумы.

Примеры

стандартный ввод	стандартный вывод
2 1 1 1 1 3 1 1	NO
2 2 1 1 1 3 1 2	YES 2 4

Замечание

В случае если разместить дракончиков по аквариумам требуемым образом нельзя, выведите единственное слово «NO». В противном случае первая строка должна содержать «YES». В следующие N строк выведите N пар чисел, задающих номера дракончиков, которых нужно поместить в двухместные аквариумы.

Первое число в паре должно описывать зелёного дракончика, второе — жёлтого. Сами пары могут идти в произвольном порядке.

Задача Н. Паросочетание максимального веса

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан двудольный граф. Количество вершин в левой и правой доле совпадает и равно n . У каждой вершины левой доли есть вес, i -й вершине соответствует вес w_i . Вес паросочетания, ребрам которого инцидентны вершины левой доли a_1, a_2, \dots, a_k есть $\sqrt{\sum_{i=1}^k w_{a_i}^2}$. Требуется найти паросочетание максимального веса.

Формат входных данных

Первая строка входного файла содержит натуральное число n — количество вершин в обеих долях ($1 \leq n \leq 1000$). Вторая строка входного файла содержит n целых чисел w_1, w_2, \dots, w_n ($1 \leq w_i \leq 1000$). Следующие n строк содержат описания ребер, инцидентных соответствующей вершине левой доли. Формат описания: количество ребер, затем номера вершин правой доли, разделенные пробелом. Суммарное количество ребер не превосходит 200000.

Формат выходных данных

Выведите n чисел — для каждой вершины левой доли выведите номер вершины правой доли, с которой ее надо взять в паросочетание. Если вершина не входит в паросочетание, выведите 0.

Примеры

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

Задача I. Футболки на олимпиаду

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Наконец-то пришло время для финала Закрытой олимпиады школьников по программированию! Все отборы пройдены, сувенирные футболки лежат в пакетах. Вот только распределение участников по площадкам еще не спланировано.

У жюри олимпиады есть m площадок проведения. На i -й площадке лежит пакет, в котором находится a_i футболок. Это значит, что на эту площадку можно зарегистрировать не более a_i участников. Всего же среди финалистов будет n человек.

Финал Закрытой олимпиады школьников по программированию проходит в городе-герое Кленинграде. Как любой другой город Флатландии, он расположен в декартовой системе координат. Жюри знает адрес каждого финалиста — координаты X_i, Y_i , а также адреса площадок — координаты x_i, y_i .

Жюри хочет так распределить участников по точкам проведения, чтобы всем участникам хватило футболок, при этом максимальное из расстояний, которое придется пройти участнику до точки проведения, было как можно меньше. Поскольку жюри олимпиады сейчас везет на точки проведения другую сувенирку, Вас попросили провести распределение участников по площадкам.

Формат входных данных

В первой строке заданы два целых числа n, m ($1 \leq n, m \leq 500$) — количество участников олимпиады и точек проведения.

В следующих n строках заданы пары целых чисел x_i, y_i ($1 \leq x_i, y_i \leq 10^6$) — координаты i -го участника олимпиады.

В следующих m строках заданы тройки целых положительных чисел X_i, Y_i, a_i ($1 \leq X_i, Y_i \leq 10^6$, $1 \leq a_i \leq n$) — координаты i -й площадки и ее вместимость.

Гарантируется, что сумма a_i не меньше n , а так же что она не превосходит 1000.

Формат выходных данных

Выведите одно число — максимальное расстояние, которое придется пройти участнику до точки проведения олимпиады при оптимальном распределении. Ответ будет считаться верным, если его абсолютная или относительная ошибка не будет превосходить 10^{-6} . Формально, пусть ваш ответ равен a , а ответ жюри — b . Ваш ответ считается правильным, если $\frac{|a-b|}{\max(1,|b|)} \leq 10^{-6}$.

Примеры

стандартный ввод	стандартный вывод
3 2 1 1 2 3 3 2 1 1 1 2 2 2	1.0000000000
3 2 100 100 101 101 102 102 101 101 2 105 105 2	4.2426406871

Замечание

В первом примере надо отправить первого участника на первую площадку, а оставшихся двух — на вторую. Тогда первый участник пройдет расстояние 0, а второй и третий — по 1.

Во втором примере надо первого и второго участников отправить на площадку 1, а третьего — на площадку 2. Тогда первый участник пройдёт расстояние 1.41421356237, второй — 0, а третий — 4.2426406871.

Система оценки

Тесты к этой задаче состоят из девяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, прохождение тестов из условия не требуется для некоторых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Дополнительные ограничения			Необх. группы	Комментарий
		n	m	a_i		
0	0	–	–	–	–	Тесты из условия.
1	10	–	–	$a_i = n$	–	
2	11	–	$m \leq 2$	–	0	
3	15	–	$m \leq 3$	–	0, 2	
4	14	$n \leq 10$	$m \leq 10$	–	0	
5	13	$n \leq 50$	$m \leq 50$	–	0, 4	
6	7	$n \leq 200$	$m \leq 200$	–	0, 4, 5	
7	9	$n \leq 300$	$m \leq 300$	–	0, 4, 5, 6	
8	10	$n \leq 400$	$m \leq 400$	–	0, 4, 5, 6, 7	Offline-проверка.
9	11	–	–	–	0–8	Offline-проверка.