

Задача А. Просто поток

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 1024 мегабайта

Дана система из узлов и труб, по которым может течь вода. Для каждой трубы известна наибольшая скорость, с которой вода может протекать через нее. Известно, что вода течет по трубам таким образом, что за единицу времени в каждый узел (за исключением двух — источника и стока) втекает ровно столько воды, сколько из него вытекает.

Ваша задача — найти наибольшее количество воды, которое за единицу времени может протекать между источником и стоком, а также скорость течения воды по каждой из труб.

Трубы являются двусторонними, то есть вода в них может течь в любом направлении. Между любой парой узлов может быть более одной трубы.

Формат входных данных

В первой строке записано натуральное число N — количество узлов в системе ($2 \leq N \leq 100$). Известно, что источник имеет номер 1, а сток номер N . Во второй строке записано натуральное M ($1 \leq M \leq 5000$) — количество труб в системе. Далее в M строках идет описание труб. Каждая труба задается тройкой целых чисел A_i, B_i, C_i , где A_i, B_i — номера узлов, которые соединяет данная труба ($A_i \neq B_i$), а C_i ($0 \leq C_i \leq 10^4$) — наибольшая допустимая скорость течения воды через данную трубу.

Формат выходных данных

В первой строке выведите наибольшее количество воды, которое протекает между источником и стоком за единицу времени. Далее выведите M строк, в каждой из которых выведите скорость течения воды по соответствующей трубе. Если направление не совпадает с порядком узлов, заданным во входных данных, то выводите скорость со знаком минус. Числа выводите с точностью 10^{-3} .

Примеры

стандартный ввод	стандартный вывод
2	4
2	1
1 2 1	-3
2 1 3	

Задача В. Максимальный поток

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.25 секунд
Ограничение по памяти: 256 мегабайт

Задан ориентированный граф, каждое ребро которого обладает целочисленной пропускной способностью. Найдите максимальный поток из вершины с номером 1 в вершину с номером n .

Формат входных данных

Первая строка входного файла содержит n и m — число вершин и ребер в графе ($2 \leq n \leq 500$, $1 \leq m \leq 10\,000$). Последующие строки описывают ребра. Каждое ребро задается тремя числами: начальная вершина ребра, конечная вершина ребра и пропускная способность ребра. Пропускные способности не превосходят 10^9 .

Формат выходных данных

Выведите величину максимального потока между вершинами 1 и n .

Пример

стандартный ввод	стандартный вывод
4 5 1 2 1 1 3 2 3 2 1 2 4 2 3 4 1	3

Задача С. Декомпозиция потока

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

Задан ориентированный граф, каждое ребро которого обладает целочисленной пропускной способностью. Найдите максимальный поток из вершины с номером 1 в вершину с номером n и постройте декомпозицию этого потока.

Формат входных данных

Первая строка входного файла содержит n и m — количество вершин и количество ребер графа ($2 \leq n \leq 500$, $1 \leq m \leq 10000$). Следующие m строк содержат по три числа: номера вершин, которые соединяет соответствующее ребро графа и его пропускную способность. Пропускные способности не превосходят 10^9 .

Формат выходных данных

В первую строку выходного файла выведите одно число — количество путей в декомпозиции максимального потока из вершины с номером 1 в вершину с номером n . Следующий строки должны содержать описания элементарных потоков, на который был разбит максимальный. Описание следует выводить в следующем формате: величина потока, количество ребер в пути, вдоль которого течет данный поток и номера ребер в этом пути. Ребра нумеруются с единицы в порядке появления во входном файле.

Пример

стандартный ввод	стандартный вывод
4 5 1 2 1 1 3 2 3 2 1 2 4 2 3 4 1	3 1 2 2 5 1 3 2 3 4 1 2 1 4

Задача D. Великая стена

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	64 мегабайта

У короля Людовика двое сыновей. Они ненавидят друг друга, и король боится, что после его смерти страна будет уничтожена страшными войнами. Поэтому Людовик решил разделить свою страну на две части, в каждой из которых будет властвовать один из его сыновей. Он посадил их на трон в города A и B , и хочет построить минимально возможное количество фрагментов стены таким образом, чтобы не существовало пути из города A в город B .

Страну, в которой властвует Людовик, можно упрощенно представить в виде прямоугольника $m \times n$. В некоторых клетках этого прямоугольника расположены горы, по остальным же можно свободно перемещаться. Кроме этого, ландшафт в некоторых клетках удобен для строительства стены, в остальных же строительство невозможно.

При поездках по стране можно перемещаться из клетки в соседнюю по стороне, только если ни одна из этих клеток не содержит горы или построенного фрагмента стены.

Формат входных данных

В первой строке входного файла содержатся числа m и n ($1 \leq m, n \leq 50$). Во второй строке заданы числа k и l , где $0 \leq k, l, k + l \leq mn - 2$, k — количество клеток, на которых расположены горы, а l — количество клеток, на которых можно строить стену. Естественно, что на горах строить стену нельзя. Следующие k строк содержат координаты клеток с горами x_i и y_i , а за ними следуют l строк, содержащие координаты клеток, на которых можно построить стену — x_j и y_j . Последние две строки содержат координаты городов A (x_A и y_A) и B (x_B и y_B) соответственно. Среди клеток, описанных в этих $k + l + 2$ строках, нет двух совпадающих. Гарантируется, что $1 \leq x_i, x_j, x_A, x_B \leq m$ и $1 \leq y_i, y_j, y_A, y_B \leq n$.

Формат выходных данных

В первой строке выходного файла должно быть выведено минимальное количество фрагментов стены F , которые необходимо построить. В последующих F строках необходимо вывести один из возможных вариантов застройки.

Если невозможно произвести требуемую застройку, то необходимо вывести в выходной файл единственное число -1 .

Пример

стандартный ввод	стандартный вывод
5 5	3
3 8	1 3
3 2	2 3
2 4	3 1
3 4	
3 1	
1 3	
2 3	
3 3	
4 3	
5 3	
1 4	
1 5	
2 1	
5 5	

Задача Е. Чокнутый профессор

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

В Университете города М. проводят эксперимент. Преподаватели сами решают, что они будут читать в рамках того или иного курса. И вот преподаватель математического анализа (в простонародье — матана) оценил по некоторым критериям все известные ему темы в данном курсе. В результате этой ревизии каждой теме сопоставлено некоторое целое число (возможно, отрицательное) — полезность данной темы. Профессор хочет максимизировать суммарную полезность прочитанных им тем, но не все так просто. Для того что бы студенты поняли некоторые темы, необходимо, чтобы были прочитаны так же некоторые другие темы, так как некоторые доказательства базируются на фактах из других тем. Однако если существует цикл из зависимостей тем, то их все можно прочитать, и на качестве понимания материала студентами это не скажется.

Вас попросили составить список тем, которые профессор должен прочитать, таким образом, чтобы студенты все поняли, и суммарная полезность курса была максимальна.

Формат входных данных

Первая строка входного файла содержит одно число — N ($1 \leq N \leq 200$). Вторая строка содержит N целых чисел, не превосходящих по модулю 1000 — полезности каждой темы. Далее следуют N строк с описанием зависимостей тем. Каждое описание начинается количеством тем, которые необходимо понять для понимания данной темы. Потом следуют номера этих тем, разделенные пробелами.

Формат выходных данных

Выведите единственное число — максимально возможную суммарную полезность прочитанного материала.

Примеры

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

Задача F. Поедание сыра

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

На сырном заводе во Флатландии живут мыши. Они очень любят сыр и часто уничтожают запасы сыра, приготовленные для отправки в магазин.

Всего на заводе живет m мышей. Для i -й мыши известна ее скорость поедания сыра s_i , мышь может съесть s_i грамм сыра в час.

Недавно мышам стал известен план работы завода на ближайшее время. Планируется изготовить n головок сыра. Про каждую головку известны r_i — к началу какого часа она будет изготовлена, d_i — в начале какого часа она начнет портиться, и p_i — вес головки сыра в граммах.

Мыши решили съесть весь сыр. В любой момент времени каждая мышь может есть некоторую головку сыра. Мыши — существа брезгливые, и одну и ту же головку сыра не могут есть одновременно несколько мышей. При этом в любой момент времени мышь может решить прекратить есть головку сыра и приняться за другую, в том числе ту, которую ранее ела другая мышь.

Мыши не любят есть сыр после того как он начал портиться. Но оставлять сыр недоеденным мыши не могут. Они решили организовать поедание сыра таким образом, чтобы величина t , такая что какую-либо головку все еще продолжают есть через t часов после того как она начала портиться, была минимальна. Помогите мышам выяснить, как это сделать.

Формат входных данных

Первая строка входного файла содержит два целых числа n и m ($1 \leq n \leq 30$, $1 \leq m \leq 30$). Следующие n строк содержит по три целых числа: p_i , r_i и d_i ($1 \leq p_i \leq 10^5$, $0 \leq r_i < d_i \leq 10^7$). Далее следуют m строк, каждая из которых содержит по одному целому числу s_j ($1 \leq s_j \leq 10^5$).

Формат выходных данных

Выведите одно вещественное число — искомое минимальное t . Ваш ответ должен отличаться от правильного не больше чем на 10^{-4} .

Примеры

стандартный ввод	стандартный вывод
2 2 13 0 4 10 1 3 4 2	0.5
1 1 1 0 1 1	0

Замечание

В первом примере мышам следует организовать поедание сыра следующим образом. Сначала первая мышь начинает есть первую головку сыра. Когда появляется вторая головка, она перестает есть первую и начинает есть вторую (в этот момент от первой осталось 9 граммов). Вторая мышь принимается есть первую головку сыра. Через 2.5 часа первая мышь доедает вторую головку сыра (на 0.5 часа позже чем она начала портиться) и снова начинает есть первую (вторая мышь за это время съела еще 5 граммов от первой головки и от нее осталось 4 грамма). Таким образом еще за час первая мышь доедает первую головку, также на 0.5 часа позже чем она начала портиться.

Во втором примере мышь успевает съесть сыр до того как он начинает портиться.

Задача G. Живопись

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

В стране Олимпия очень развита живопись. Картиной считается любой прямоугольник, который состоит из черных и белых единичных квадратов. Художник Олимпус решил радикально улучшить свои картины. Для этого он планирует к белому и черному цветам добавить еще и серый оттенок. По его задумке, граница между каждым черным и белым квадратом должна содержать серую линию, чтобы образовался эффект плавного перехода.

Однако, перед началом работы, он обнаружил, что серая краска очень дорого стоит. Чтобы сэкономить деньги художник решил оценить, не выгоднее ли сначала перекрасить некоторые белые квадраты в черные, а черные в белые для того, чтобы минимизировать расходы на краску.

Напишите программу, которая по информации о существующей картине определяет минимальную сумму денег, которые понадобятся на ее улучшение.

Формат входных данных

Первая строка входного файла содержит пять натуральных чисел N , M , w , b , g . ($1 \leq N, M \leq 70$) — высота и ширина картины, ($1 \leq w, b, g \leq 1000$) — цена рисования одного белого единичного квадрата, черного единичного квадрата и серой линии единичной длины, соответственно.

Далее следует N строк, каждая из которых состоит из M литер. Литера B соответствует черному квадрату, а W — белому.

Формат выходных данных

Единственная строка выходного файла должна содержать одно целое число, которое есть минимальной суммой затрат на улучшение картины.

Пример

стандартный ввод	стандартный вывод
3 2 10 12 1 BW WB BW	7

Задача Н. Нефтепроводы

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

Олигарх Вован, как и большинство других олигархов, занимается транспортировкой нефти из Западной Кукуляндии в Восточную Кукуляндию. В его владении находится огромная нефтедобывающая станция в Западной Кукуляндии, не меньшего размера нефтеперерабатывающая станция в Восточной Кукуляндии, а также система нефтепроводов, по которым нефть перегоняется из одной страны в другую. На столе у Вована лежит карта нефтепроводов. Хотелось бы знать, какое количество условных единиц нефти может транспортировать данная система. Каждый нефтепровод соединяет некоторую пару станций. На карте все станции пронумерованы, при этом добывающая станция имеет номер 1, перерабатывающая — номер N , а транзитные — номера от 2 до $N - 1$. Каждый нефтепровод может транспортировать ограниченное количество нефти, зато в любом направлении. Вован не знает, что Земля круглая, поэтому каждая станция на его карте имеет плоские координаты (x_i и y_i — координаты i -й станции). Нефтепроводы являются отрезками прямых. На карте пара нефтепроводов может пересекаться только по общей станции-вершине. Известно, что среди всех станций добывающая станция имеет наименьшую координату x , а перерабатывающая — наибольшую координату x .

Формат входных данных

В первой строке дано целое число N — количество станций на карте ($2 \leq N \leq 10000$). В следующих N строках перечислены координаты станций (x_i, y_i) через пробел. Координаты — целые числа, по модулю не превосходящие 10^8 . В следующей строке дано целое число M — количество нефтепроводов. Далее в M строках через пробел перечислены характеристики нефтепроводов — пара номеров станций, соединяемых нефтепроводом, а также пропускная способность нефтепровода в условных единицах — целое число от 1 до 10^8 . Гарантируется, что система нефтепроводов может транспортировать некоторое ненулевое количество нефти, но не может транспортировать более $2 \cdot 10^9$ условных единиц нефти.

Формат выходных данных

В первой строке выведите наибольшее количество условных единиц нефти, которое может транспортировать система Вована. В следующих M строках выведите план транспортировки — тройки чисел (A, B, C), означающие, что из станции A в станцию B должно течь C условных единиц нефти. Все нефтепроводы должны быть представлены в данном списке ровно один раз (даже те, по которым ничего не течёт). Число C во всех тройках должно быть неотрицательным.

Пример

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

Задача I. Максимальный поток минимальной стоимости

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

Задан ориентированный граф, каждое ребро которого обладает пропускной способностью и стоимостью. Найдите максимальный поток минимальной стоимости из вершины с номером 1 в вершину с номером n .

Формат входных данных

Первая строка входного файла содержит n и m — количество вершин и количество ребер графа ($2 \leq n \leq 100$, $0 \leq m \leq 1000$). Следующие m строк содержат по четыре целых числа: номера вершин, которые соединяет соответствующее ребро графа, его пропускную способность и его стоимость. Пропускные способности и стоимости неотрицательны и не превосходят 10^5 .

Формат выходных данных

В выходной файл выведите одно число — цену максимального потока минимальной стоимости из вершины с номером 1 в вершину с номером n . Ответ не превышает $2^{63} - 1$.

Пример

стандартный ввод	стандартный вывод
4 5 1 2 1 2 1 3 2 2 3 2 1 1 2 4 2 1 3 4 2 3	12

Задача J. Расшифровка

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

Недавно на уроке во время контрольной Мария Ивановна перехватила записку от Саши к Оле. Мария Ивановна очень хочет знать, что в записке, но, к сожалению, записка зашифрована. Мария Ивановна знает, что её ученики для шифровки заменяют каждую букву исходного сообщения на какую-то другую. Замена происходит таким образом, что одинаковые буквы всегда заменяются одной и той же буквой, а разные — разными.

Мария Ивановна подозревает, что записка — это ответы к контрольному тесту (ведь её длина случайно оказалась равной длине строки с правильными ответами). Однако она знает, что ответы Саши не обязательно полностью правильны. На каждый вопрос возможен один из K вариантов ответа. Естественно, Мария Ивановна знает правильные ответы.

Мария Ивановна решила расшифровать записку таким способом, чтобы максимизировать количество правильных ответов Саши. Однако, она очень занята, поэтому попросила Вас помочь ей в этом пустяковом деле.

Формат входных данных

В первой строке задана длина каждой из строк N ($1 \leq N \leq 2\,000\,000$) и K — количество возможных ответов на каждый вопрос ($1 \leq K \leq 52$). Ответы нумеруются в порядке `abcde...xyzABCDE...XYZ`. То есть, при $K = 6$ возможные ответы выглядят как `abcdef`, а при $K = 30$ — `abcde...xyzABCD`.

Во второй строке задана зашифрованная записка — строка, состоящая из строчных и заглавных латинских букв.

В третьей строке заданы правильные ответы — строка той же длины, что и первая, состоящая из строчных и заглавных латинских букв.

Формат выходных данных

В первой строке выведите единственное число — максимально возможное количество правильных ответов у Саши.

Во второй строке выведите расшифровку — строчку длины K , где по порядку для каждой буквы из шифра учеников указано, какому ответу она соответствует.

Если несколько расшифровок дают правильный ответ, выведите любую.

Примеры

стандартный ввод	стандартный вывод
10 2 aaabbbbaab bbbbabbbbb	7 ba
10 2 aaaaaaabbb bbbbaabbb	6 ab
9 4 dacbdacbd acbdacbdba	9 cdba

Задача К. Автоматное программирование

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

В один замечательный день в компанию «Х» завезли k автоматов. И не простых автоматов, а автоматов-программистов! Это был последний неудачный шаг перед переходом на андроидов-программистов, но это уже совсем другая история.

В компании сейчас n задач, для каждой из которых известно время начала ее выполнения s_i , длительность ее выполнения t_i и прибыль компании от ее завершения c_i . Любой автомат может выполнять любую задачу, ровно одну в один момент времени. Если автомат начал выполнять задачу, то он занят все моменты времени с s_i по $s_i + t_i - 1$ включительно и не может переключиться на другую задачу.

Вам требуется выбрать набор задач, которые можно выполнить с помощью этих k автоматов и который принесет максимальную суммарную прибыль.

Формат входных данных

В первой строке записаны два целых числа n и k ($1 \leq n \leq 1000$, $1 \leq k \leq 50$) — количество задач и количество автоматов, соответственно.

В следующих n строках через пробелы записаны тройки целых чисел s_i, t_i, c_i ($1 \leq s_i, t_i \leq 10^9$, $1 \leq c_i \leq 10^6$), s_i — время начала выполнения i -го задания, t_i — длительность i -го задания, а c_i — прибыль от его выполнения.

Формат выходных данных

Выведите n целых чисел $x_1, x_2, \dots, x_n$. Число x_i должно быть равно 1, если задачу i следует выполнить, и 0 в противном случае.

Если оптимальных решений несколько, то выведите любое из них.

Примеры

стандартный ввод	стандартный вывод
3 1 2 7 5 1 3 3 4 1 3	0 1 1
5 2 1 5 4 1 4 5 1 3 2 4 1 2 5 6 1	1 1 0 0 1

Задача L. Поток в двудольном графе

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

Дан двудольный граф, исток и сток. Каждая доля состоит из N вершин. Из истока в i -ю вершину левой доли ведёт ребро пропускной способности a_i . Из i -й вершины правой доли ведёт ребро в сток пропускной способности b_i . Также между некоторыми вершинами левой и правой доли есть двусторонние ребра бесконечной пропускной способности.

Вам требуется найти величину максимального потока из истока в сток.

Формат входных данных

В первой строке заданы два числа N и M ($1 \leq N \leq 10^4$, $0 \leq M \leq 10^5$) — число вершин в каждой доле и число рёбер соответственно.

Во второй строке записаны N чисел a_i ($1 \leq a_i \leq 10^4$).

Во третьей строке записаны N чисел b_i ($1 \leq b_i \leq 10^4$).

В каждой из последующих M строк записаны пары чисел u, v ($1 \leq u, v \leq n$) означающие наличие ребра между вершиной u левой доли и вершиной v правой доли.

Формат выходных данных

Выведите одно число — величину максимального потока.

Пример

стандартный ввод	стандартный вывод
3 4 3 2 1 5 4 4 1 1 1 2 2 3 3 3	6

Задача М. Пение

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

Алиса и Боб собираются спеть песню вместе. Вам даны три целых числа: n , low и $high$. Для простоты, обозначим числами от 1 до n высоты нот, которые встречаются в песне (от самых низких до самых высоких). Алиса может петь только ноты с высотами от low до n включительно. Боб может петь только ноты с высотами от 1 до $high$ включительно. (Гарантируется, что эти отрезки пересекаются).

Вам дана последовательность высот нот в песне. Каждая нота в песне должна быть спета ровно одним человеком: или Алисой, или Бобом. Все ноты одной высоты должны быть спеты одним человеком. Количество раз, которое сменяется певец, в течении песни должно быть минимальным возможным.

Формально, смена певца происходит в тот момент, когда одну ноту пел один человек, а следующую — уже другой.

Вычислите минимально возможное количество смен певца в течении песни.

Формат входных данных

В первой строке даны четыре целых числа n , low , $high$ и m ($1 \leq low \leq high \leq n \leq 1000$, $1 \leq m \leq 1000$).

Во второй строке даны m чисел — последовательность нот из песни h_i ($1 \leq h_i \leq n$).

Формат выходных данных

Выведите единственное число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
3 2 2 6 1 2 3 2 1 2	2
10 3 7 8 4 4 5 5 6 5 3 6	0
6 2 5 6 5 3 1 6 4 2	1
10 4 5 9 1 4 3 5 2 5 7 5 9	3
100 20 80 24 2 27 3 53 53 52 52 60 85 89 100 53 60 2 3 53 100 89 40 42 2 53 2 85	5

Задача N. Плотник

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

Такеша — известный плотник, он принимает много заказов со всей Японии. Однако недавно воры украли у него все инструменты, и теперь ему придется купить новые, чтобы выполнять заказы.

В магазине инструментов Такеша узнал цены на каждый из m интересующих его инструментов, кроме того, он выяснил, что на некоторые пары инструментов есть скидка, если их покупать вместе. При этом каждый инструмент встречается не более чем в одной такой паре.

Сейчас у Такеша есть n заказов, для каждого заказа он знает его стоимость и список инструментов, которые нужны, чтобы его выполнить.

Такеша необязательно выполнять все заказы. Он хочет выбрать какое-то множество заказов и купить только те инструменты, которые нужны для их выполнения. При этом он хочет максимизировать свою прибыль, то есть разницу между стоимостью выполненных заказов и стоимостью инструментов с учетом скидок. Помогите Такеше сделать это.

Формат входных данных

Первая строка ввода содержит числа n , m и p — число заказов, число инструментов и число скидок ($1 \leq n \leq 100$, $2 \leq m \leq 100$, $1 \leq p \leq m/2$).

Следующие n строк содержат описание заказов. Каждое описание начинается со стоимости заказа (положительное целое число не больше 1000), далее идет число инструментов, необходимых для его выполнения и номера этих инструментов.

Следующие m строк содержат стоимости инструментов без скидки (положительные целые числа не больше 1000).

Следующие p строк содержат описание скидок. Каждое описание содержит номера инструментов и их суммарную стоимость с учетом скидки. Гарантируется, что каждый инструмент встречается только в одной скидке. Также гарантируется, что сумма с учетом скидки больше стоимости каждого из инструментов по отдельности и меньше их суммы.

Формат выходных данных

Выведите максимальную прибыль, которую может получить Такеша.

Пример

стандартный ввод	стандартный вывод
3 4 2 100 2 1 2 100 1 3 100 1 4 20 20 50 150 1 2 30 3 4 180	120

Задача О. Высокий маразм

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

Город Нью-Васюки представляет собой таблицу N на N кварталов. Таким образом, в нем существует $(N + 1)(N + 1)$ перекрестков и $2N(N + 1)$ двусторонних дорог. Каждый перекресток имеет высоту. Известно, что левый верхний перекресток имеет высоту 0, а правый нижний – высоту 1. Для каждой дороги известно, сколько человек идет в каждом направлении по этой дороге. При этом, если дорога ведет от перекрестка i к перекрестку j , и разность высот $h = h_j - h_i$, то неудобство перемещения для каждого человека равно $\max(h, 0)$.

Для всех клеток, кроме двух угловых вы вправе выбирать любую высоту. Найдите распределение, при котором суммарное неудобство будет минимальным.

Формат входных данных

В первой строке входного файла содержится число N – размер таблицы, $1 \leq N \leq 500$. Далее следует $4N(N+1)$ чисел – искомое число людей для каждой дороги в естественном порядке. Сначала следует $N(N + 1)$ чисел для каждой дороги с запада на восток, затем столько же чисел с севера на юг, потом столько же чисел с востока на запад, а затем столько же чисел с юга на север. Для каждого направления дороги перечислены с севера на юг, а затем – с запада на восток. Обратите внимание на пример (там все числа специально различны).

Все количества являются целыми неотрицательными числами не превосходящими миллиона.

Формат выходных данных

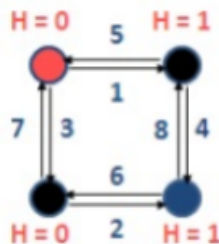
Выведите одно число – итоговое минимальное суммарное неудобство.

Пример

стандартный ввод	стандартный вывод
1 1 2 3 4 5 6 7 8	3

Замечание

Высоты могут быть произвольными вещественными числами! Картинка к примеру из условия:



Задача Р. Две перестановки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Даны две перестановки $p_0, p_1, \dots, p_{n-1}$ и $q_0, q_1, \dots, q_{n-1}$ чисел $0, 1, \dots, n-1$. Надо построить две перестановки A и B чисел $0, 1, \dots, n-1$, удовлетворяющие следующим условиям:

- Для каждого i ($0 \leq i < n$) A_i должно быть равно i или p_i .
- Для каждого i ($0 \leq i < n$) B_i должно быть равно i или q_i .

Расстоянием между перестановками A и B назовём количество индексов i , что $A_i \neq B_i$. Найдите максимальное возможное расстояние между A и B .

Формат входных данных

Первая строка содержит единственное число n ($1 \leq n \leq 10^5$) — размер перестановок.

Вторая строка содержит n чисел $p_0, p_1, \dots, p_{n-1}$ ($0 \leq p_i < n$) — первая перестановка.

Третья строка содержит n чисел $q_0, q_1, \dots, q_{n-1}$ ($0 \leq q_i < n$) — вторая перестановка.

Формат выходных данных

Выведите максимальное возможное расстояние между A и B .

Примеры

стандартный ввод	стандартный вывод
4 2 1 3 0 0 2 3 1	3
10 0 4 5 3 7 8 2 1 9 6 3 8 5 6 4 0 2 1 7 9	8

Задача Q. Колдуны

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

Есть n колдунов, пронумерованных от 1 до n . i -й колдун имеет силу a_i и планирует победить монстра с силой b_i .

Можно сколько угодно раз делать следующую операцию:

- Увеличить силу любого колдуна на 1. То есть выбрать любое i от 1 до n и заменить a_i на $a_i + 1$.

Пара колдунов (i, j) считается хорошей, если выполнено хотя бы одно из двух условий:

- $a_i \geq b_i$ и $a_j \geq b_j$.
- $a_i \geq b_j$ и $a_j \geq b_i$.

Даны m пар колдунов (x_i, y_i) . Определите минимальное количество операций, чтобы сделать каждую пару хорошей.

Формат входных данных

Первая строка содержит единственное число n ($1 \leq n \leq 100$) — количество колдунов.

Дальше идут n строк. i -я из них содержит два числа a_i и b_i ($1 \leq a_i, b_i \leq 100$) — параметры i -го колдуна.

Следующая строка содержит единственное число m ($1 \leq m \leq \frac{n(n-1)}{2}$) — количество пар колдунов.

Дальше идут m строк. i -я строка содержит два числа x_i и y_i ($1 \leq x_i < y_i \leq n$) — пара колдунов, которую надо сделать хорошей.

Гарантируется, что все пары различны.

Формат выходных данных

Выведите минимальное число операций, чтобы сделать каждую пару хорошей.

Примеры

стандартный ввод	стандартный вывод
5 1 5 2 4 3 3 4 2 5 1 3 1 4 2 5 3 5	2
4 1 1 1 1 1 1 1 1 3 1 2 2 3 3 4	0
9 1 1 2 4 5 5 7 10 9 3 9 13 10 9 3 9 2 9 7 1 5 2 5 1 6 2 4 3 4 4 9 8 9	22

Задача Р. Мадока и первая сессия

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

О нет, на первой же сессии Мадоке попался билет со следующей сложной задачей:

Дано число n и m пар чисел (v_i, u_i) . А также есть массив $b_1, b_2, \dots, b_n$, **изначально заполненный нулями**.

Затем для каждого индекса i , где $1 \leq i \leq m$, выполняется либо $b_{v_i} := b_{v_i} - 1$ и $b_{u_i} := b_{u_i} + 1$, либо $b_{v_i} := b_{v_i} + 1$ и $b_{u_i} := b_{u_i} - 1$. Обратите внимание, что ровно одна из этих операций должна быть выполнена для каждого i .

Также дан массив s размера n , состоящий только из 0 и 1. И массив $a_1, a_2, \dots, a_n$, где гарантируется, что если $s_i = 0$, то $a_i = 0$.

Помогите Мадоке и определите, можно ли выполнить операции выше таким образом, чтобы для каждого i , где $s_i = 1$, выполнялось $a_i = b_i$. И если возможно, то как это сделать.

Формат входных данных

Первая строка содержит два целых числа n и m ($2 \leq n \leq 100000, 1 \leq m \leq 100000$) — длина массива a и количество пар чисел.

Вторая строка содержит n целых чисел $s_1, s_2, \dots, s_n$ ($0 \leq s_i \leq 1$) — элементы массива s .

Третья строка содержит n целых чисел $a_1, a_2, \dots, a_n$ ($|a_i| \leq m$) — элементы массива a . Гарантируется, что если $s_i = 0$, то $a_i = 0$.

i -я из следующих m строк содержат два целых числа v_i и u_i ($1 \leq v_i, u_i \leq n, v_i \neq u_i$) — индексы элементов массива b , к которым применяется операция. Также гарантируется, что не существует таких двух индексов i и j , где $1 \leq i < j \leq m$, что $(v_i, u_i) = (v_j, u_j)$ или $(v_i, u_i) = (u_j, v_j)$.

Формат выходных данных

Выведите в первой строке «YES», если можно выполнить операции нужным образом, и «NO» в противном случае.

Вы можете выводить каждую букву в любом регистре (например, «YES», «Yes», «yes», «yEs» будут распознаны как положительный ответ).

В случае, если вы вывели «YES», выведите m пар целых чисел. Если для пары (v_i, u_i) нужно выполнить $b_{v_i} := b_{v_i} - 1$ и $b_{u_i} := b_{u_i} + 1$, выведите (v_i, u_i) . Иначе выведите (u_i, v_i) . Если существует несколько способов получить правильный ответ, можно вывести любой из них.

Пары можно выводить в любом порядке.

Примеры

стандартный ввод	стандартный вывод
5 5 1 1 1 1 1 -2 0 2 1 -1 1 5 1 4 3 5 3 4 4 5	YES 1 5 1 4 5 3 4 3 5 4
5 5 0 1 0 1 0 0 1 0 0 0 1 3 2 3 3 5 3 4 4 5	YES 3 1 3 2 5 3 3 4 4 5
4 4 1 1 1 1 0 2 -2 2 1 3 1 4 2 3 2 4	NO

Замечание

В первом примере массив b будет меняться следующим образом:
 $[0, 0, 0, 0, 0] \rightarrow [-1, 0, 0, 1, 0] \rightarrow [-2, 0, 0, 1, 1] \rightarrow [-2, 0, 1, 0, 1] \rightarrow [-2, 0, 2, 0, 0] \rightarrow [-2, 0, 2, 1, -1]$.
 $a_i = b_i$ для всех индексов i от 1 до 5.

Во втором примере нам достаточно, чтобы в конце $b_2 = 1$, поскольку только $s_2 = 1$.

В третьем примере входных данных нельзя выполнить операции нужным образом.

Задача S. Ошибка туристического оператора

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



В задаче тесты разделены на группы. Ограничения указаны для полной задачи, оцениваемой в 10 баллов.

Путешествовать в одиночку скучно, поэтому вы собрали туристическую группу из n_1 мальчиков и n_2 девочек (мальчики пронумерованы от 1 до n_1 , а девочки от 1 до n_2) и поехали в Турцию. В Ваши планы не входило лежать на пляже, поэтому первым делом вы обратились к туристическому оператору и попросили организовать экскурсии для всех.

Ваша группа получила k различных достопримечательностей, пронумерованных от 1 до k . Каждый член группы выписал те, куда он хотел бы поехать. Таким образом, i -й мальчик хочет посетить s_i достопримечательностей с номерами $a_{i,1} < a_{i,2} < \dots < a_{i,s_i}$. Другие достопримечательности он посещать не будет. Аналогично, j -я девочка хочет посетить t_j достопримечательностей с номерами $b_{j,1} < b_{j,2} < \dots < b_{j,t_j}$.

Для посещения достопримечательности с номером i требуется заплатить c_i турецких лир.

Предположим, что i -й мальчик или девочка посетят достопримечательности с суммарной стоимостью билетов равной S . Тогда его (или её) **недовольство** в этом случае будет равно S^2 . Общее **недовольство** вашей группы увеличивается на **недовольство** каждого её члена.

Не все члены вашей группы знают друг друга, но они очень хотят завести новые знакомства. Так сложилось, что все мальчики уже знают друг друга, и все девочки тоже знают друг друга. Потому новые знакомства возможны только между мальчиком и девочкой. Известно, что всего есть m пар людей v_i и u_i ($1 \leq v_i \leq n_1$, $1 \leq u_i \leq n_2$), которые не знают друг друга. **Удовольствие** от нового знакомства между мальчиком v_i и девочкой u_i равно w_i , поэтому общее **недовольство** уменьшится на w_i за каждую достопримечательность, которую они посетят вместе.

Туристический оператор, которого вы выбрали, очень странный. Его сотрудник узнал ваши предпочтения и хотел составить хорошую программу, но ошибся и **максимизировал** общее **недовольство** вашей туристической группы. Найдите **недовольство** вашей группы, которое у него получилось.

После этого Вы, конечно, сможете составить себе план самостоятельно и получить максимум удовольствия!

Формат входных данных

Первая строка входных данных содержит четыре целых числа n_1 , n_2 , k и m ($1 \leq n_1, n_2, k \leq 50$, $1 \leq m \leq 300$) — количество мальчиков, девочек, достопримечательностей и пар людей, которые хотят познакомиться, соответственно.

Вторая строка входных данных содержит k целых чисел $c_1, c_2, \dots, c_k$ ($1 \leq c_i \leq 10^4$) — стоимости посещения достопримечательностей.

Каждая из следующих m строк содержит три целых числа v_i , u_i и w_i ($1 \leq v_i \leq n_1$, $1 \leq u_i \leq n_2$, $1 \leq w_i \leq 10^9$) — мальчик и девочка, которые хотят познакомиться друг с другом, и **удовольствие** от их знакомства, соответственно. Гарантируется, что все пары (v_i, u_i) различны.

Каждая из следующих n_1 строк содержит описание достопримечательностей, которые могут посетить мальчики. i -я из них сначала содержит число s_i ($1 \leq s_i \leq k$), а затем s_i целых чисел $a_{i,1} < a_{i,2} < \dots < a_{i,s_i}$ ($1 \leq a_{i,j} \leq k$) — номера достопримечательностей, которые может посетить i -й мальчик.

Каждая из следующих n_2 строк содержит описание достопримечательностей, которые могут посетить девочки. j -я из них сначала содержит число t_j ($1 \leq t_j \leq k$), а затем t_j целых чисел $b_{j,1} < b_{j,2} < \dots < b_{j,t_j}$ ($1 \leq b_{j,j} \leq k$) — номера достопримечательностей, которые может посетить j -я девочка.

Гарантируется, что $\sum s_i + \sum t_i \leq 400$.

Формат выходных данных

Выведите единственное целое число — **максимально недовольство** от плана туристического оператора.

Система оценки

Задача имеет 4 группы тестов. Итоговый балл выставляется по сумме баллов за пройденные группы.

1. Группа тестов, имеющая дополнительное ограничение $n_1 = n_2 = 1$, оценивается в 2 балла.
2. Группа тестов, имеющая дополнительное ограничение $k = 1$, оценивается в 2 балла.
3. Группа тестов, имеющая дополнительное ограничение $k = 2$, оценивается в 3 балла.
4. Группа тестов, соответствующая оригинальным ограничениям, оценивается в 3 балла. Для получения баллов за неё необходимо пройти все предыдущие группы и примеры.

Примеры

стандартный ввод	стандартный вывод
3 3 5 3 1 2 3 4 5 1 1 31 2 2 32 3 3 33 4 1 2 3 4 4 1 2 3 4 4 1 2 3 4 4 1 2 3 5 4 1 2 3 5 4 1 2 3 5	415
2 3 6 5 1 4 4 1 1 4 2 3 26 1 3 172 2 1 96 1 2 51 1 1 31 2 2 5 6 1 2 3 4 5 6 4 1 4 5 6 1 5 5 1 2 3 4 6	292