

## 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. Улиточки

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 1024 мегабайта

Две улиточки Маша и Петя сейчас находятся в на лужайке с абрикосами и хотят добраться до своего домика. Лужайки пронумерованы числами от 1 до  $n$  и соединены дорожками (может быть несколько дорожек соединяющих две лужайки, могут быть дорожки, соединяющие лужайку с собой же). По соображениям гигиены, если по дорожке проползла улиточка, то вторая по той же дорожке уже ползти не может. Помогите Пете и Маше добраться до домика.

### Формат входных данных

В первой строке файла записаны четыре целых числа —  $n$ ,  $m$ ,  $s$  и  $t$  (количество лужаек, количество дорог, номер лужайки с абрикосами и номер домика). В следующих  $m$  строках записаны пары чисел. Пара чисел  $(x, y)$  означает, что есть дорожка с лужайки  $x$  до лужайки  $y$  (из-за особенностей улиток и местности дорожки односторонние). Ограничения:  $2 \leq n \leq 10^5$ ,  $0 \leq m \leq 10^5$ ,  $s \neq t$ .

### Формат выходных данных

Если существует решение, то выведите YES и на двух отдельных строчках сначала последовательность лужаек для Машеньки (дам нужно пропускать вперед), затем путь для Пети. Если решения не существует, выведите NO. Если решений несколько, выведите любое.

### Примеры

стандартный ввод	стандартный вывод
3 3 1 3	YES
1 2	1 3
1 3	1 2 3
2 3	

### Замечание

Дан оргграф, найти два непересекающихся по ребрам пути из  $s$  в  $t$ , вывести вершины найденных путей.

## Задача D. Разрез

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 мегабайта

Найдите минимальный разрез между вершинами 1 и  $n$  в заданном неориентированном графе.

### Формат входных данных

На первой строке входного файла содержится  $n$  ( $2 \leq n \leq 100$ ) — число вершин в графе и  $m$  ( $0 \leq m \leq 400$ ) — количество ребер. На следующих  $m$  строках входного файла содержится описание ребер. Ребро описывается номерами вершин, которые оно соединяет, и его пропускной способностью (положительное целое число, не превосходящее 10 000 000), при этом никакие две вершины не соединяются более чем одним ребром.

### Формат выходных данных

На первой строке выходного файла должны содержаться количество ребер в минимальном разрезе и их суммарная пропускная способность. На следующей строке выведите возрастающую последовательность номеров ребер (ребра нумеруются в том порядке, в каком они были заданы во входном файле).

### Примеры

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

## Задача F. 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	3
1 2 1	1 2 2 5
1 3 2	1 3 2 3 4
3 2 1	1 2 1 4
2 4 2	
3 4 1	

## Задача G. 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 двунаправленные 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	6
3 2 1	
5 4 4	
1 1	
1 2	
2 3	
3 3	

## Задача Н. Великая стена

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 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  $n$  ( $1 \leq m, n \leq 50$ ). Следующие  $m$  строк по  $n$  символов задают карту страны. Символы обозначают: «#» — гора, «.» — место, пригодное для постройки стены, «-» — место, не пригодное для постройки стены, «A» и «B» — города  $A$  и  $B$ .

### Формат выходных данных

В первой строке выходного файла должно быть выведено минимальное количество фрагментов стены  $F$ , которые необходимо построить. Далее нужно вывести карту в том же формате, как во входном файле. Клетки со стеной обозначьте символом «+».

Если невозможно произвести требуемую застройку, то выведите в выходной файл единственное число  $-1$ .

### Примеры

стандартный ввод	стандартный вывод
5 5 --... A-.#- .##- --- --.-B	3 --+. . A-+#- +##- --- --.-B
1 2 AB	-1
2 2 A# #B	0 A# #B

## Задача I. Живопись

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

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, которая понадобится на улучшение картины.

### 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, являющееся минимальной суммой затрат на улучшение картины.

### Пример

<code>drawing.in</code>	<code>drawing.out</code>
3 2 10 12 1 BW WB BW	7



## Задача J. Плотник

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

Такеши — 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 Такеши 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 Такеши есть  $n$  заказов, для каждого заказа он знает его стоимость и список инструментов, которые нужны, чтобы его выполнить.

Такеши 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 Такеши 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 Такеши.

### 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 К. 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 L. 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.