

Задача А. 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 В. Улиточки

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

Две улиточки Маша и Петя сейчас находятся 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). Ввиду соображений 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 , a и h (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: $2 \leq n \leq 10^5, 0 \leq m \leq 10^5, a \neq h$

Формат выходных данных

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

Пример

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

Задача С. 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 D. Химия!!!

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Вася и Сережа играют в следующую игру. В некоторых клетках клетчатого листка Сережа рисует один из символов 'H', 'O', 'N' или 'C', после чего Вася должен провести между некоторыми находящимися в соседних клетках символами линии так, чтобы получилось корректное изображение химической молекулы. К сожалению, Сережа любит рисовать много символов, и Вася не может сразу определить, возможно ли вообще нарисовать линии нужным способом. Помогите ему написать программу, которая даст ответ на этот вопрос.

В этой задаче проведенные между символами химических элементов линии будем считать корректным изображением молекулы, если они удовлетворяют следующим условиям:

- каждая линия соединяет символы, нарисованные в соседних (по стороне) клетках,
- между каждой парой символов проведено не более одной линии,
- от каждого элемента отходит ровно столько линий, какова валентность этого элемента (1 для H, 2 для O, 3 для N, 4 для C),
- пустые клетки ни с чем не соединены, и
- хотя бы в одной клетке нарисован какой-то символ.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m ($1 \leq n, m \leq 50$) — размеры листочка, на котором рисует Сережа. Далее следуют n строк по m символов в каждой, задающих конфигурацию химических элементов, которую нарисовал Сережа; пустые клетки задаются символом '.'.

Формат выходных данных

В выходной файл выведите одно слово: 'Valid', если линии провести требуемым образом можно, и 'Invalid', если нельзя.

Примеры

стандартный ввод	стандартный вывод
3 4 HOH. NSOH OO..	Valid
3 4 HOH. NSOH OONH	Invalid

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. Альянсы

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

В сказочном мире есть большой остров прямоугольной формы. Стороны острова, равны r миль и s миль, и весь остров разделен на $r \times s$ квадратных участков. Некоторые участки необитаемы, а остальные занимают деревни фантастических существ: эльфов, людей, гномов и хоббитов. Каждый участок содержит не более одной деревни. Две деревни считаются соседними, если их участки имеют общую сторону.

В последнее время деревни напуганы Великим Злом. Чтобы чувствовать себя безопаснее, каждая деревня решила сформировать военные альянсы с некоторыми из соседей. Альянс всегда включает в себя две соседние деревни, и это взаимное и симметричное соглашение.

В зависимости от расы существ, проживающих в деревне, должны выполняться различные свойства, чтобы жители чувствовали себя в безопасности:

- Эльфы чувствуют себя уверенно и поэтому нуждаются в одном альянсе.
- Люди должны заключить альянс ровно с двумя соседями. Более того, оставлять две противоположные стороны деревни подвергнутыми воздействию плохо по тактическим соображениям. Поэтому два соседа, с которыми заключили союз, не могут быть расположены по разные стороны от деревни.
- Гномы должны заключить альянс ровно с тремя соседями.
- Хоббиты крайне пугливы и поэтому должны заключить альянс со всеми четырьмя соседями.

Другими словами, каждая деревня кроме людей должна заключить определенное количество альянсов, не важно с какими из соседей. Для людей есть дополнительное ограничение: союзные деревни не должны находиться с противоположных сторон.

Условия должны выполняться независимо от местонахождения деревни на карте. Например, пусть есть деревня гномов, ей нужно заключить три альянса. Если она расположена на побережье, это означает, что она должна заключить союз со всеми тремя соседями. А если она расположена в углу, то ее жители никогда не будут чувствовать себя в безопасности.

Формулировка задачи

Дано описание острова, ваша задача — решить, можно ли создать альянсы так, чтобы все жители острова чувствовали себя в безопасности. В случае положительного ответа ваша программа должна предложить одну из возможных конфигураций альянсов. В случае нескольких решений выберите произвольное.

Формат входных данных

Первая строка ввода содержит два целых числа r и s , определяющие размер острова ($1 \leq r, s \leq 70$). Следующие r строк задают описание острова. Каждая строка состоит из s чисел, разделенных пробелами:

- 0 означает необитаемый участок,
- 1 означает деревню эльфов,
- 2 означает деревню людей,
- 3 означает деревню гномов.
- 4 означает деревню хоббитов,

(Обратите внимание, что число на входе всегда соответствует количеству желаемых альянсов для этой деревни.)

Формат выходных данных

Если решения нет, выведите одну строку «Impossible!». В противном случае выведите какой-то вариант карты альянсов в следующем формате.

Каждая область должна отображаться на выходе в виде матрицы 3×3 из символов. Если область необитаема, соответствующая секция должна быть заполнена точками. Если в этой области есть деревня, в середине должен быть символ «O» (большая латинская буква O), представляющий саму деревню, и должны быть символы «X» (большая латинская буква X), соединяющие ее с союзниками. Остальная часть матрицы должна быть заполнена точками.

Примеры

стандартный ввод	стандартный вывод
3 4 2 3 2 0 3 4 3 0 2 3 3 1OXXOXXO.... .X..X..X.... .X..X..X.... .OXXOXXO.... .X..X..X.... .X..X..X.... .OXXOXXOXXO.
1 2 2 1	Impossible!

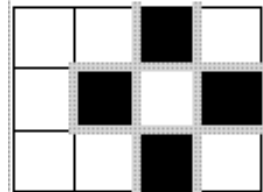
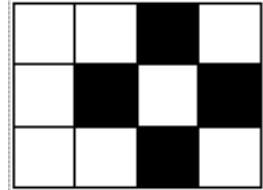
Задача Н. Живопись

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 В 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 I. Разбиение на пары

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

Космические археологи обнаружили на планете в соседней звездной системе n древних артефактов, которые они пронумеровали от 1 до n . Каждый артефакт имеет k различных параметров, каждый параметр характеризуется целым числом. Артефакт i имеет параметры $a_{i,1}, a_{i,2}, \dots, a_{i,k}$. Оказалось, что первые параметры у всех артефактов различны: для всех $i \neq j$ выполнено $a_{i,1} \neq a_{j,1}$, при этом другие параметры у артефактов могут совпадать.

Учёные также обнаружили текст, в соответствии с которым для активации артефактов их необходимо особым образом разбить на пары и совместить. Разбиение артефактов на пары является корректным, если для каждого t от 1 до k можно выбрать такое число b_t , что оно лежит на отрезке между значениями t -го параметра артефактов каждой пары. То есть, если артефакты i и j образуют пару, должно выполняться условие $a_{i,t} \leq b_t \leq a_{j,t}$ или условие $a_{i,t} \geq b_t \geq a_{j,t}$.

Теперь ученые хотят выяснить, верно ли расшифрован текст. Для этого необходимо проверить, существует ли корректное разбиение артефактов на пары. Каждый артефакт должен войти ровно в одну пару в разбиении.

Требуется написать программу, которая по описанию параметров артефактов определяет, можно ли разбить их на пары таким образом, чтобы для каждого параметра существовало значение, лежащее между значениями этого параметра артефактов каждой пары, и в случае положительного ответа выводит такое разбиение.

Формат входных данных

В первой строке заданы целые числа n и k — количество артефактов и количество параметров ($2 \leq n \leq 2 \cdot 10^5$, n чётно, $1 \leq k \leq 7$).

В следующих n строках задано по k целых чисел $a_{i,1}, a_{i,2}, \dots, a_{i,k}$ — параметры артефактов ($-10^9 \leq a_{i,j} \leq 10^9$, все значения $a_{i,1}$ различны).

Формат выходных данных

Выведите «NO», если требуемого разбиения на пары не существует.

В противном случае выведите «YES» в первой строке. Далее выведите $n/2$ строк, в каждой строке выведите по два числа — номера артефактов, из которых следует составить пару. Каждый артефакт должен быть выведен ровно один раз. Если существует несколько корректных разбиений артефактов на пары, разрешается вывести любое из них.

Примеры

стандартный ввод	стандартный вывод
6 2 8 6 1 5 6 3 3 1 4 7 7 2	YES 4 1 5 6 2 3
4 3 1 -1 -1 2 1 1 3 -1 1 4 1 -1	NO