

Задача A. Crisis

Имя входного файла: `crisis.in`
Имя выходного файла: `crisis.out`
Ограничение по времени: 12 seconds
Ограничение по памяти: 64 Mebibytes

Во время экономического кризиса курс акций одной крупной IT-компании вёл себя совершенно нетривиально: в первый день кризиса он увеличился на некоторое целое число, во второй — уменьшился, также на целое число, в третий — снова увеличился и так далее.

Оказалось, что в каждый нечётный день курс увеличивается на число из некоторого множества A , а в каждый чётный — уменьшается на число из другого множества B .

По заданным множествам A и B выясните, сможет ли курс на некоторый день стать в точности равным любому наперёд заданному целому числу?

Формат входного файла

Во входном файле задано целое число T ($1 \leq T \leq 500$) — количество тестовых примеров. Каждый тестовый пример начинается со строки, в которой заданы целые числа n и m ($1 \leq m, n \leq 10^4$) — мощность множеств A и B соответственно. Далее следуют n целых положительных чисел, не превосходящих 10^9 — элементы множества A , и m целых положительных чисел в том же диапазоне — элементы множества B .

Формат выходного файла

Для каждого тестового примера выведите 'YES', если курс на некоторый день сможет стать в точности равным любому наперёд заданному целому числу, и 'NO' в противном случае.

Пример

<code>crisis.in</code>	<code>crisis.out</code>
2	NO
1 1	YES
3	
3	
2 2	
1	
3	
1	
3	

Задача В. Server safety

Имя входного файла: `debug.in`
Имя выходного файла: `debug.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

В одной крупной IT-компании за время новогодних праздников случился сбой сервера. Хотя дублирующие мощности и справились с нагрузкой, в первый рабочий день руководство технического отдела в срочном порядке потребовало предоставить варианты решений, которые исключат подобные поломки в дальнейшем.

Сервер состоит из нескольких модулей, некоторые из которых зависят друг от друга. Доступ к серверу осуществляется через модуль n . Некоторые модули могут функционировать некорректно, что приводит к некорректному функционированию зависящих от них модулей. Для каждого модуля i известна его устойчивость t_i : если менее t_i модулей, от которых зависит модуль i , функционирует некорректно, то модуль i функционирует корректно ($t_i = 0$ обозначает, что модуль i сам по себе может функционировать некорректно). При этом зависимость модулей друг от друга не может быть циклической.

Для каждого модуля i производитель может установить систему безопасности, которая стоит p_i . При установке такой системы модуль i функционирует корректно вне зависимости от функционирования других модулей. Требуется найти наименьшую сумму, которую нужно затратить, чтобы добиться бесперебойной работы модуля n .

Формат входного файла

В первой строке входного файла задано целое число n ($1 \leq n \leq 20$) — количество модулей. Далее следуют n строк — описания модулей. В i -й такой строке сначала следует целое число — устойчивость i -го модуля t_i ($0 \leq t_i \leq n$), затем цена системы безопасности p_i для этого модуля, затем идёт список (возможно, пустой) номеров $m_i < i$ модулей, от которых зависит работа модуля i . Модули нумеруются в порядке их перечисления во входном файле с единицы, номера модулей в списке не повторяются.

Формат выходного файла

В выходной файл выведите с точностью 0.01 минимальную стоимость защиты, гарантирующей бесперебойную работу модуля n .

Пример

debug.in	debug.out
7	721.70
0 1744.0	
1 210.5 1	
1 313.1	
1 4817.2 2 3	
4 3922 1 2 3 4	
0 511.2 1 3 5	
1 71482 2 5 6	

Задача C. Investments

Имя входного файла: `invest.in`
Имя выходного файла: `invest.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Крупная IT-компания собирается открыть целевые программы в одном или нескольких вузах. У руководства компании есть информация по $1 \leq n \leq 100$ ведущим вузам страны. Также эксперты выяснили, какой доход принесёт сотрудничество с каждым из n вузов при выделении вузам k грантов на развитие (где k пробегает значение от 1 до некоторого числа m — максимального количества грантов, которое можно выделить одному вузу). При этом ни для какого вуза большее количество полученных грантов не может привести к уменьшению дохода от сотрудничества с ним. Учитывая, что всего выделено $0 \leq B \leq 100$ грантов, требуется найти максимальный доход IT-компания в результате открытия целевых программ.

Формат входного файла

В первой строке входного файла заданы три целых числа n , m и B ($1 \leq n, m, B \leq 100$) — количество вузов в списке, максимальное количество грантов, выделяемое одному вузу и общее количество грантов.

Далее заданы n строк, каждая из которых состоит из $m + 1$ чисел, заданных не более, чем с 2 знаками после десятичной точки. i -е число в j -й строке задаёт доход от сотрудничества с i -м вузом при условии выделения тому $j - 1$ грантов.

Формат выходного файла

Выведите одно число с точностью 0.01 — максимальный доход, который IT-компания сможет получить от сотрудничества с вузами в результате открытия целевых программ.

Пример

<code>invest.in</code>	<code>invest.out</code>
4 5 7 0 3.3 3.3 3.3 3.3 3.3 0 0 0 0 4.1 0 2.0 3 4.0 5.0 6 0 0 3.3 3.3 3.3 4.9	11.60

Задача D. Мар

Имя входного файла: `map.in`
Имя выходного файла: `map.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Одна крупная IT-компания начала разработку глобального картографического проекта «Экология океана».

Этот проект, в частности, по спутниковым снимкам вычисляет место загрязнения океана, после чего моделирует распространение загрязнения, используя данные о циркулирующих в море течениях.

Ваша задача — составить по карте течений и точке первоначального загрязнения карту загрязнённости прямоугольного участка океана через несколько дней после загрязнения. Участок разбит на единичные квадраты, в одном из которых и произошло загрязнение.

Формат входного файла

В первой строке заданы 5 целых чисел x , y , x_0 , y_0 , t ($1 \leq x, y \leq 100$, $1 \leq x_0 \leq x$, $1 \leq y_0 \leq y$, $0 \leq t \leq 100$) — соответственно высота и ширина карты, координаты x и y квадрата первоначального загрязнения и время, на которое надо выдать прогноз. В последующих y строках задана карта течений. Каждая строка состоит из $2x$ символов, первый из которых — число от 0 до 4, задающее силу течения, а второй — один из символов 'N', 'E', 'S', 'W' — направление течения. Течение силы k в загрязнённом квадрате приводит к тому, что на следующий день k соседних в направлении, совпадающем с направлением течения, квадратов также оказываются загрязнены.

Формат выходного файла

Выведите y строк по x символов в каждой — карту соответствующего участка океана на t -й день после первоначального загрязнения. Незагрязнённый квадрат отмечается точкой '.', загрязнённый — символом 'X'.

Пример

map.in	map.out
14 8 8 4 6	XXXXXXXXX.....
2W3S2W4S4W2S2W1W2S1E2S3E2S3E	.XXX.XXX.....
0E1E0S2N0S2E1E1N0E0W3S0N0E0S	.XXX.X.XX.....
0E1E0S2N0S0E2E1E4S0W2S0N0E0S	XXXX...XXXXXXXX
4W1W4W1W4W1W4W4N4E4E4E4E4E	..XX...XXXXXX
4W1W4W1W4W1W4W1W3E3E3E3E3EXXXXXX
4W3W3W3W3W3W3W2E2S2E2N2E2E	..XXXXXXXXXXXX
4W1W4W1W4W1W4W1W1W4E4E4E4EXX...
0W0W1W1W0E1S1W1W0N1E1W1E1W4E	

Задача E. Ecosystem of Aquarium

Имя входного файла: `sysaqua.in`
Имя выходного файла: `sysaqua.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

В рамках проекта «Экология» планируется написание средств моделирования различных экосистем. Вам досталась простейшая из этих задач — промоделировать экосистему аквариума.

Изначально аквариум пуст, затем в него запускают планктон, затем по очереди начинают запускать голодных рыбок разных видов. Первый вид рыбок питается исключительно планктоном, а каждый следующий вид рыбок питается некоторыми видами рыбок, запущенными в аквариум до этого, и только ими. Попав в аквариум, рыбки тут же начинают питаться. Если какой-то рыбке не хватает её полного рациона (к примеру, представительнице вида 3 необходимы 2 рыбки вида 1 и 1 рыбка вида 2, а в аквариуме плавают всего 1 рыбка вида 1 и 2 рыбки вида 2), она погибает, при этом погибшие рыбки съедобными не являются. Следующий вид запускается уже после того, как все выжившие представители предыдущего вида наелись. Ваша задача — определить количество живых представителей каждого вида в аквариуме после того, как представители последнего запущенного туда вида по той или иной причине перестанут питаться.

Формат входного файла

В первой строке входного файла задано число n ($1 \leq n \leq 100$) — общее количество видов (включая планктон). i -я из следующих n строк имеет следующий формат: целое неотрицательное число p_i ($0 \leq p_i \leq 1000$) — количество особей вида i , запущенных в аквариум, после чего заданы $i - 1$ чисел e_j ($0 \leq e_j \leq 10$) — полный рацион вида i (то есть количество особей видов $1, \dots, i - 1$, которые необходимо съесть одной особи вида i , чтобы выжить).

Формат выходного файла

В каждой из n строк выходного файла выведите количество особей каждого вида, оставшихся в аквариуме после завершения эксперимента.

Пример

<code>sysaqua.in</code>	<code>sysaqua.out</code>
6	69
220	0
30 3	0
1020 0 5	6
310 1 0 1	11
22 2 0 0 0	11
11 1 0 0 0 1	

Задача F. Yet Another Fish Problem

Имя входного файла: `yafp.in`
Имя выходного файла: `yafp.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Одним из пользователей системы «Экология» является экипаж рыбного траулера «Nordic Victory», занимающегося добычей рыбы в Балтийском море.

Капитан траулера с помощью системы обнаружил в районе действия сети траулера N крупных рыб, каждая из которых плывёт с неизменной скоростью в неизменном направлении. Сеть представляет собой окружность, центром которой является траулер (для данной задачи размерами траулера можно пренебречь).

Траулер должен выловить не менее K рыб, при этом капитан, чтобы не нанести ущерба природе, хочет использовать сеть как можно меньшего радиуса. Заметим, что на траулерах типа «Nordic Victory» сеть может быть в любой момент мгновенно установлена на нужный радиус.

Ваша задача — написать программу, вычисляющую требуемый радиус сети.

Формат входного файла

В первой строке входного файла заданы два целых числа C_x и C_y ($1 \leq C_x, C_y \leq 10^4$) — координаты траулера. Вторая строка содержит два целых числа N и K ($5 \leq N \leq 1000$, $1 \leq K \leq N$) — общее количество обнаруженных рыб и минимальное количество рыб, которое должно оказаться в сети. Каждая из последующих N строк содержит 4 целых числа P_x, P_y, D_x, D_y — соответственно x - и y - координаты рыбы в начальный момент и в момент $t = 1$ ($0 \leq P_x, P_y, D_x, D_y \leq 10^4$). Заметим, что траулер может использовать сеть в любой (не обязательно целый) момент времени, начиная с времени 0.

Формат выходного файла

В выходной файл выведите одно число — минимальный радиус сети, при выборе которого в оптимальном случае будет поймано не менее K рыб. Ответ выводить с точностью до 10^{-5} .

Пример

<code>yafp.in</code>	<code>yafp.out</code>
7 8 5 4 2 5 2 12 5 13 11 11 5 10 5 8 3 5 11 5 4 2 10 2	5.000000