Задача A. Three dices

Имя входного файла:	`dices.in`
Имя выходного файла:	`dices.out`
Ограничение по времени:	6 seconds
Ограничение по памяти:	64 Mebibytes

В новой MMORPG «Tenth Century — Magic Time» (TCMTime) для определения стартовых характеристик персонажа введено следующее правило. Игрок n раз бросает три стандартных игральных кости с 6 гранями. Получившиеся числа, называемые *значениями бросков* (n целых чисел в диапазоне между 1 и 18) сортируются по неубыванию. Для генерации параметров персонажа используются числа, начиная с i-го и заканчивая j-м. Так, например, при $n = 10$, $i = 1$, $j = 5$ используются 5 наихудших из 10 бросков, при $n = 10$, $i = 6$, $j = 10$ используются 5 наилучших из 10 бросков.

По заданным n, i и j вычислите математическое ожидание значения одного броска, используемого для генерации персонажа.

Формат входного файла

В единственной строке входного файла заданы три целых числа n, i и j — соответственно количество бросков, нижняя и верхняя границы диапазона ($1 \leqslant i \leqslant j \leqslant n \leqslant 500$).

Формат выходного файла

Выведите одно число — значение требуемого математического ожидания с точностью 10^{-5}.

Примеры

dices.in	dices.out
10 1 5	8.28868884
10 6 10	12.71131116
2 1 1	8.82368827
10 1 10	10.5

Задача B. Maze with division

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

На одном из островов недавно появившегося в мире TCMTime континента есть замечательный лабиринт. Он представляет собой квадрат $N \times N$, состоящий из целых положительных чисел. Перед входом в лабиринт, находящимся в левом верхнем углу квадрата, написано «входное» число I, после выхода, находящегося в правом нижнем углу квадрата, написано «выходное» число O. Чтобы пройти лабиринт, персонаж должен стартовать с «входного» числа и войти в лабиринт. Далее он может двигаться прямо, поворачивать направо или налево (ходы назад запрещены!), но только в случае, если сумма чисел на текущем и на предыдущем шаге делится на число, написанное в том квадрате, куда планируется ход. Персонаж может посещать одну клетку неограниченное количество раз.

По заданному лабиринту требуется выяснить, сможет ли персонаж достичь «выходного» числа O (хотя оно и расположено вне лабиринта, но выход на него осуществляется по тем же правилам, по которым осуществляется передвижение по лабиринту), и, если сможет, привести последовательность прохождения лабиринта, состоящую из наименьшего возможного числа шагов. Гарантируется, что минимальная последовательность единственна.

Формат входного файла

В первой строке входного файла заданы три целых числа — N, I и O ($1 \leqslant N \leqslant 10, 1 \leqslant I, O \leqslant 2^{20}$). Далее следует описание лабиринта — N строк по N целых чисел a_{ij} в каждой ($1 \leqslant a_{ij} \leqslant 2^{20}$).

Формат выходного файла

Выведите требуемую в условии задачи последовательность чисел, начинающуюся с I и заканчивающуюся O, в порядке их прохождения персонажем по пути от входа к выходу. В случае, если пройти лабиринт насквозь нельзя, выведите "`Impossible`".

Примеры

divmaze.in	divmaze.out
3 1 5 3 4 7 4 3 11 7 1 4	1 3 4 7 1 4 5
2 8 3 16 2 8 3	Impossible

Задача C. Fraction

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

У персонажей онлайн-игры TCMTime текущие характеристики могут быть и дробными. При этом обычно они выводятся в виде (возможно, периодических) десятичных дробей. Но некоторые игроки легче воспринимают — как ни странно — обычные дроби. В связи с этим вам поручили написать модуль, преобразующий десятичные дроби в вид X/Y.

Задана десятичная дробь в виде $I.N$, $I.(P)$ или $I.N(P)$, где I — целая часть дроби, N — непериодическая часть дроби, P — период дроби. Требуется представить дробь в виде X/Y, где X — целое неотрицательное число, Y — целое положительное число, взаимно простое с X (в случае $X \neq 0$).

Формат входного файла

Десятичная дробь в виде $I.N$, $I.(P)$ или $I.N(P)$, где I — целая часть дроби, N — непериодическая часть дроби, Z — один из её периодов. Общее количество цифр в записи дроби не превосходит восьми.

Формат выходного файла

Выведите заданную дробь в виде "`X / Y`" (с пробелами перед и после знака дроби).

Примеры

fraction.in	fraction.out
0.50	1 / 2
1.(33)	4 / 3

Задача D. Magic Wood

Имя входного файла:	`magicwood.in`
Имя выходного файла:	`magicwood.out`
Ограничение по времени:	6 seconds
Ограничение по памяти:	64 Mebibytes

В игре TCMTime есть магический лес, деревья в котором рождаются и гибнут по правилам, аналогичным правилам игры «Жизнь» Конвея. Лес представляет собой прямоугольник $M \times N$ клеток.

Каждую секунду конфигурация леса меняется по следующим правилам:

- Соседними с данной клеткой считаются все клетки, имеющие с ней хотя бы одну общую точку. У клетки, расположенной не на границе леса, 8 соседних клеток.
- Если в клетке растёт дерево, а в соседних с ним клетках растёт менее n_{min} или более n_{max} деревьев, то на следующую секунду дерево погибает (и клетка становится свободной).
- Если клетка свободна, а в соседних с ней клетках растут более p деревьев, то на следующую секунду в этой клетке рождается дерево.
- Остальные клетки остаются в том же состоянии, в каком были в текущую секунду.

Введём понятие *стартовой* конфигурации. *Стартовой* считается такая конфигурация магического леса, которая не может быть получена из какой-либо конфигурации (включая и саму себя) по вышеперечисленным правилам. Так что стартовая конфигурация может быть получена только сразу после инициализации данного участка игры.

По заданной конфигурации требуется вычислить, может ли она быть получена из какой-либо стартовой конфигурации за конечное количество секунд, и, если может, вывести минимальное количество секунд, за которое это могло произойти.

Формат входного файла

В первой строке входного файла заданы четыре целых числа — количество строк M, количество столбцов N, а также параметры n_{min}, n_{max}, p ($1 \leqslant M \leqslant 4$, $1 \leqslant N \leqslant 5$, $1 \leqslant n_{min} < n_{max} \leqslant 8$, $1 \leqslant p \leqslant 8$). Далее следуют M строк по N символов в каждой — описание текущей конфигурации магического леса. Если на пересечении i-го столбца и j-й строки находится символ '.', то данная клетка свободна, если же там находится символ '*', то в этой клетке находится дерево.

Формат выходного файла

Выведите одно число — минимальное количество секунд, за которое заданная конфигурация могла быть получена из некоторой стартовой, или -1, если заданная конфигурация не могла быть получена ни из какой стартовой конфигурации.

Примеры

magicwood.in	magicwood.out
2 2 1 2 2	1
4 5 2 3 2 **.*. *..*. .**..	2
4 5 2 3 2 **.*. ..*.* ***.. ....*	0
4 4 2 3 2 *.** **.* *.** **.*	-1

Задача E. Pet Bugs

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

В игре TCMTime персонажи, принадлежащие к магической школе природы (Nature), могут использовать во время поединка прирученных животных, в том числе и насекомых. При этом управлять прирученным насекомым игрок может с помощью системы команд. Для совершения атаки насекомое должно находиться от атакуемого монстра на некотором расстоянии. Поэтому во время сражения возможность определить минимальное расстояние между насекомым, движущимся по заданной игроком траектории, и целью оказывается весьма полезной.

Насекомые (в отличие от обычных монстров и персонажей) перемещаются в трёхмерном пространстве. Игрок может дать насекомому команду пролететь вперёд некоторое количество метров, повернуть направо, налево, вверх или вниз относительно текущего направления движения.

По заданным начальным координатам насекомого, последовательности команд, отправленных игроком и координатам цели вычислите, на каком минимальном расстояние от цели находилось насекомое.

Формат входного файла

В первой строке входного файла заданы три целых числа b_x, b_y и b_z — начальные координаты насекомого. Первоначально насекомое ориентировано так, что направление движения совпадает с положительным направлением оси x, вверх от насекомого ведёт положительное направление оси z, а влево — положительное направление оси y.-

Во второй строке заданы три целых числа t_x, t_y и t_z — координаты цели ($-10^4 \leqslant b_x, b_y, b_z, t_x, t_y, t_z \leqslant 10^4$).

В последующих строках задана последовательность команд. i-я строка описывает i-ю команду и состоит из двух чисел d_i r_i, где d_i — расстояние, которое пролетит насекомое в текущем направлении движения, а r_i равно -1 в случае, если эта команда завершает последовательность, 0 для поворота направо, 90 для поворота вверх, 180 для поворота налево и 270 для поворота вниз. При этом насекомое сначала пролетает расстояние d_i, находясь в текущей ориентации, а уже потом выполняет поворот (или заканчивает движение). Количество команд не превышает 100, $0 \leqslant d_i \leqslant 100$.

Формат выходного файла

Выведите одно число — минимальное расстояние, на котором насекомое находилось от цели во время движения по заданной траектории, с точностью до 10^{-2}.

Пример

petbug.in	petbug.out
0 0 0 2 2 2 4 180 4 180 4 90 4 90 4 180 4 270 4 0 2 -1	2.83
0 0 0 1 1 1 1 180 1 90 6 -1	0

Задача F. Race and Class

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

В игре TCMTime у каждого персонажа есть два параметра — раса (например, Human, Gnome, Elf) и класс (например, Warrior, Cleric, Druid, Illusionist), определённые однозначно на момент генерации персонажа. Для каждой расы и каждого класса у вас есть информация о том, сколько на данный момент зарегистрировано персонажей данной расы (данного класса).

Ваша задача — найти такое распределение персонажей по парам "раса/класс", которое соответствовало бы имеющейся информации. При этом известно, что некоторые расы несовместимы с некоторыми классами (то есть число персонажей с такой парой "раса/класс" заведомо равно нулю).

Формат входного файла

В первой строке входного файла заданы два целых числа M и N — общее количество рас и классов в игре, соответственно ($1 \leqslant M, N \leqslant 10$). Далее следуют M строк, описывающих расы. Строка начинается целым числом m_i, обозначающих количество персонажей, имеющих данную расу, после которого через пробел задано название расы — непустую строку длиной не более 64 символов, содержащую только строчные и прописные латинские буквы, а также пробелы. При этом название не может начинаться или заканчиваться на пробел, а также содержать более двух пробелов подряд. В последующих N строках аналогичным образом описываются классы. Далее следует строка, содержащая одно целое число K ($1 \leqslant K \leqslant MN$) — количество пар "раса-класс", допустимых в данной игре. В последних K строках входного файла заданы сами допустимые пары в порядке "раса/класс". Название расы и класса разделено символом "/".

Формат выходного файла

Для каждой заданной во входном файле пары "раса/класс" выведите её название в том же виде, в котором оно задано во входном файле, после чего через пробел выведите количество персонажей с данными параметрами. Если возможны несколько вариантов распределения, удовлетворяющих требованию задачи, выведите любой.

Пример

raceclass.in	raceclass.out
2 2 8 Gnome 100 Drow 90 Warrior 18 Wizard 3 Gnome/Warrior Drow/Warrior Drow/Wizard	Drow/Warrior: 82 Gnome/Warrior: 8 Drow/Wizard: 18