

Во ВСЕХ ЗАДАЧАХ ВВОД ИЗ STANDARD INPUT, ВЫВОД В STANDARD OUTPUT!

Задача А. Тонкости конспирации (TIME LIMIT 2s, MEMORY LIMIT 64M)

После победы над англичанами в последнем матче отборочного турнира (победа, которая обеспечила сборной России выход на Евро2008), сборная Хорватии стала одной из самых любимых команд россиян. Письма с благодарностями и подарки буквально завалили посольство Хорватии в России. Надо ли говорить, что при виде любого хорвата каждый российский болельщик считает своим долгом выразить ему свою симпатию! Поэтому тренерский штаб Хорватии всерьез опасается срыва подготовительного процесса из-за чрезмерного внимания к сборной со стороны благодарных поклонников из России.

Футболисты сборной Хорватии расположатся в Австрии в одном из лучших отелей. Одним из основных условий выбора отеля были гарантии того, что футбольные болельщики и назойливые журналисты не смогут проникать в отель и отвлекать команду от процесса подготовки к играм.

Кроме того, были приняты и дополнительные меры. Чтобы футболистов труднее было найти в большом помещении отеля, каждый из них должен строго один раз в час перемещаться из одного помещения отеля в другое. Причем, нужно не просто переходить из номера в номер. Неплохо было бы задействовать и другие помещения – кафе, душевые, бар и т.д.

Поклонники хорватской сборной впали в полное отчаяние, ведь принятые меры эффективно прячут футболистов от посторонних. И они обратились к вам за помощью с просьбой определить возможное местоположение спортсмена после N перемещений для каждого помещения отеля.

ВХОДНЫЕ ДАННЫЕ

В первой строке записано два положительных целых числа N ($N < 10^9$) и M ($M < 10000$). Далее следует M строк, каждая из которых описывает наличие возможности перехода между двумя помещениями отеля. Строка состоит из двух названий помещений, разделенных ровно одним пробелом. Каждое помещение упоминается во входных данных хотя бы один раз. Названия помещений состоят из латинских букв и имеют длину не более 10 символов. Общее число различных помещений, представленных во входных данных, не превосходит 100. Переходы между помещениями являются однонаправленными.

ВЫХОДНЫЕ ДАННЫЕ

Для каждого помещения необходимо вывести список помещений, в которых можно оказаться после N перемещений, стартуя из этого помещения.. Каждая строка вывода должна иметь формат:

`<стартовое помещение>: <помещение 1> <помещение 2> ... <помещение K>`

В каждой строке вида `<помещение 1> <помещение 2> ... <помещение K>` элементы необходимо упорядочить по возрастанию в соответствии с критерием упорядочивания. В свою очередь строки вида `<стартовое помещение>: <помещение 1> <помещение 2> ... <помещение K>` должны быть упорядочены по возрастанию по строке `<стартовое помещение>` в соответствии с критерием упорядочивания.

КРИТЕРИЙ УПОРЯДОЧИВАНИЯ

Из двух строк меньше та, длина которой меньше. Если длины строк равны, меньше та, первый символ которой имеет меньший ASCII-код. Если же первые символы равны, сравниваются вторые и т.д., пока не будут найдены два различных символа.

Пример входных данных	Пример выходных данных
-----------------------	------------------------

3 8

roomA roomB

roomB bar

bar roomA

cafe roomA

cafe roomB

roomA shower

Hiddink shower

roomB cafe

bar: bar cafe

cafe: bar cafe roomA roomB

roomA: roomA roomB

roomB: bar cafe roomB shower

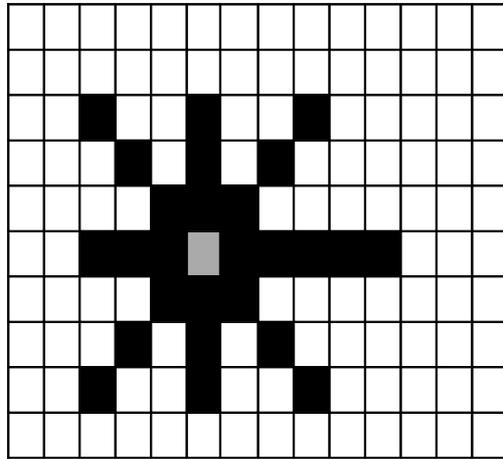
shower:

Hiddink:

Задача В. РИСУНКИ НА ТАБЛО (TIME LIMIT 2s, MEMORY LIMIT 64M)

Европейский футбольный союз (УЕФА) и организаторы чемпионата Европы по футболу будут стараться из всех сил, чтобы Евро2008 запомнилось как самое необычное (конечно, оно таким и будет, если, например, сборная России выиграет трофей. Но поскольку организаторы уже далеко не маленькие дети, то в чудеса давно не верят. Поэтому им необходимо придумать другие способы сделать чемпионат запоминающимся).

Руководитель УЕФА, великий французский футболист 80-х годов прошлого века Мишель Platini предложил развлекать зрителей на стадионе в перерыве футбольного матча. Для этого каждый футбольный стадион Евро2008 оснащен двумя электронными табло размером $M_1 \times N_1$ и $M_2 \times N_2$ клеток. В перерыве матча на обоих табло отображаются картинки в виде снежинки. Снежинка имеет восемь лучей, однако длина лучей не всегда одинакова. Угол между соседними лучами всегда одинаковый. На рисунке представлена снежинка, семь лучей которой имеют длину 3 клетки, а один луч – 5 клеток. Центральная клетка снежинки (на рисунке выделена диагональной штриховкой) не относится ни к одному из лучей.



Зрителям предлагается определить, являются ли две снежинки на разных табло условно равными. Снежинки считаются условно равными, если из одной из них (любой) можно получить другую путем поворота (возможно, на нулевой угол) и масштабирования (возможно, на коэффициент 1). Поворачивать разрешается на угол, кратный 90 градусов. При масштабировании изменяются лишь длины лучей, ширина лучей размер центра снежинки не меняется.

Мишель Platini очень надеется, что его идея окажется удачной, но для этого необходимо написать программу для быстрого сравнения снежинок. А кто может справиться с такой сложной проблемой в такие сжатые сроки лучше российских программистов?

ВХОДНЫЕ ДАННЫЕ

В первой строке записаны два целых числа M_1 и N_1 . Далее идет изображение первого табло. Затем записаны два числа M_2 и N_2 ($3 \leq M_1, N_1, M_2, N_2 \leq 1000$). Затем записано изображение второго табло. Пустая клетка табло обозначается точкой. Клетка табло, принадлежащая снежинке, обозначается #. Длина луча снежинки может составлять от 1 до 499 единиц.

ВЫХОДНЫЕ ДАННЫЕ

Необходимо вывести "Not equal", если снежинки не являются условно равными. Иначе в первой строке вывести "Equal", во второй строке вывести абсолютное значение угла поворота и коэффициент масштабирования. Коэффициент масштабирования должен быть целым числом, не

меньшим 1. Если возможно несколько правильных ответов, выведите ответ с минимальным абсолютным значением угла поворота.

Пример входных данных	Пример выходных данных
<pre>8 12#.#..... ...###..... ..#####..... ...###..... ..#.#.#..... 10 13#...#...#... ..#...#...#... ...#.#.#.....###..... .#####.....###..... ...#.#.#..... ..#...#..... .#...#.....</pre>	<pre>Equal 90 2</pre>

Задача С. ГЕНЕРАТОР РИСУНКОВ (TIME LIMIT 2s, MEMORY LIMIT 64M)

Первоначально разработку программ для рисования снежинок на новых табло стадионов поручили английским программистам. Чиновники УЕФА и организаторы Евро2008 от Австрии и Швейцарии несколько раз встречались с исполнителями заказа, чтобы проинспектировать состояние дел. Вначале дело спорилось. Однако, как только стало известно, что сборная Англии не попала на чемпионат Европы, энтузиазм английских программистов сошел на нет.

И вот уже организаторы чемпионата Европы очень озабочены ситуацией со зрительским конкурсом в перерыве футбольных матчей. Ведь без специальной программы, рисующей снежинку на табло по ее описанию, проведение зрительского конкурса в перерыве матча будет невозможным!

Программа должна заполнять все табло точками, а затем чертить снежинку с заданными длинами лучей. Центр снежинки должен располагаться в точке в точке (X, Y) . Клетка в левом верхнем углу табло имеет координаты $(0,0)$.

Организаторы Евро2008 готовы наградить автора программы билетом на один из матчей с участием сборной России.

ВХОДНЫЕ ДАННЫЕ

Первая строка ввода содержит два целых числа N и M – размеры табло ($3 \leq N, M \leq 1000$). Во второй строке записаны два целых числа X и Y ($1 \leq X, Y \leq 998$). Далее записан квадрат из чисел с длиной стороны 3 – три строки по три целых числа через пробел. Этот квадрат задает длины лучей (от 1 до 499) в соответствующих направлениях. Центральный элемент квадрата всегда равен 0.

ВЫХОДНЫЕ ДАННЫЕ

Выведите N строк по M символов – табло с рисунком снежинки на нем. Если снежинка с заданными параметрами целиком не умещается на табло, выведите `Invalid parameters`

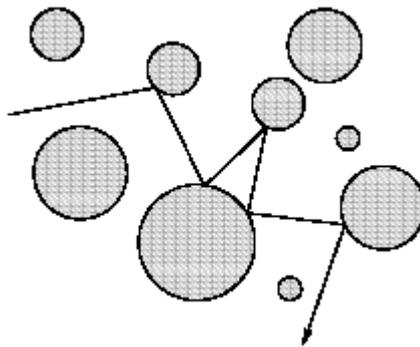
Пример входных данных 10 12 5 6 2 3 3 4 0 3 3 2 3	Пример выходных данных#..#..#.#.#...###.... ..#####..###....#.#.#... ...#.....#..
Пример входных данных 10 12 5 6 2 3 8 4 0 3 3 2 3	Invalid parameters

ЗАДАЧА D. ОТРАЖЕНИЯ (TIME LIMIT 2s, MEMORY LIMIT 64M)

К Евро2008 компания SportsEA решила выпустить новый футбольный симулятор. Для построения реалистичных изображений объектов в нем используется метод трассировки лучей.

Для рендеринга изображения с использованием трассировки лучей вычисляется путь, который проходит луч света в сцене. Попробуйте написать программу, которая вычисляет такой путь в ограниченном пространстве. Для простоты будем рассматривать двумерную сцену. Все объекты в сцене – зеркальные (полностью отражают падающий на них луч) сферы. Когда луч света падает на сферу, он отражается таким образом, что угол падения w относительно касательной в точке падения равен углу отражения относительно той же касательной.

Следующий рисунок демонстрирует типичный путь, который световой луч может пройти в сцене:



Ваша задача – написать программу, которая для заданного описания сцены и исходной точки падения луча в сцену, определяет, на какие сферы он упадет.

ВХОДНЫЕ ДАННЫЕ

Первая строка входного файла содержит целое число n ($n \leq 15$) – количество сфер в сцене. Следующие n строк содержат по 3 целых числа x_i , y_i и r_i , разделенных одним пробелом, где (x_i, y_i) – центр i -й сферы, ($-1000 \leq x_i, y_i \leq 1000$), а r_i – ее радиус ($0 < r_i \leq 1500$). Затем следует строка, содержащая 4 целых числа x , y , d_x , d_y , описывающих световой луч ($-1000 \leq x, y \leq 1000$; $-100 \leq d_x, d_y \leq 100$). Луч начинает "движение" из точки (x, y) и первоначально движется в направлении (d_x, d_y) . По крайней мере одно из значений d_x или d_y не равно нулю.

Сферы не пересекаются и не касаются друг друга. Луч не начинается внутри сферы и никогда не падает на сферу по касательной.

ВЫХОДНЫЕ ДАННЫЕ

Выходной файл должен содержать номера сфер, от которых отражается луч (в порядке их обхода лучом), разделенные одним пробелом.

Если луч падает не более, чем на 10 сфер (и затем отражается в бесконечность), после номера последней сферы необходимо вывести "inf" (Infinity – бесконечность). Если луч падает на большее число сфер, чем десять, то необходимо вывести "... " после номера десятой сферы.

Пример входных данных #1	Пример выходных данных #1
3	1 2 1 3 inf
3 3 2	
7 7 1	
8 1 1	
3 8 1 -4	

Пример входных данных #2

```
2
0 0 1
5 0 2
2 0 1 0
```

Пример выходных данных #2

```
2 1 2 1 2 1 2 1 2 1 ...
```

Задача Е. ОПТИМАЛЬНАЯ ПРОГРАММА (TIME LIMIT 2s, MEMORY LIMIT 64M)

Организаторы Евро2008 заметили, что программа генерации рисунков на табло (см. задачу С) работает не так быстро, как хотелось бы. При детальном анализе выяснилось, что некоторые части кода вызываются огромное количество раз. Необходимо срочно оптимизировать эти части кода.

В этой задаче вам предстоит написать программу автоматической генерации оптимального ассемблерного кода. Для заданной функции (серии пар аргумент-значение) необходимо вывести содержащую минимальное количество команд программу.

Сгенерированные вами программы будут запускаться на машине с памятью в виде стека. Эта машина поддерживает пять ассемблерных команд: ADD, SUB, MUL, DIV и DUP. Первые четыре команды извлекают из стека два верхних элемента, производят соответствующую операцию и помещают результат в стек (см. рисунок). Команда DUP помещает в стек копию верхнего элемента.

Начальное состояние	ADD	SUB	MUL	DIV	DUP
					a
a					a
b	a+b	b-a	a*b	b/a	b
c	c	c	c	c	c
...
...

Здесь b/a – целочисленное деление.

В начале работы стек содержит единственное число – аргумент функции из входных данных. После окончания вычислений в стеке должно находиться единственное число – требуемое значение функции.

Существует 3 случая, при которых машина переходит в состояние ошибки:

- Деление на 0 при исполнении команды DIV.
- При выполнении бинарных операций в стеке присутствует лишь один элемент.
- Результатом операции является число, абсолютное значение которого превышает 30000.

ВХОДНЫЕ ДАННЫЕ

Входной файл содержит описание функции. В первой строке записано целое положительное число n ($n \leq 10$) – количество пар аргумент/значение. Во второй строке записано n различных целых чисел – аргументы функции. Во третьей строке записано n целых чисел – соответствующие значения функции. Все числа не превышают 30000 по абсолютному значению.

ВЫХОДНЫЕ ДАННЫЕ

Выходной файл должен содержать программу с минимальным количеством команд, вычисляющую заданную функцию без перехода в состояние ошибки. При этом программа должна содержать не более, чем 10 команд. Если возможно несколько оптимальных программ, выведите лексикографически меньшую.

Если не существует такой программы не длинее 10 команд, правильно вычисляющей функцию без перехода в состояние ошибки, выведите `Impossible` Если оптимальная программа не содержит команд, выведите `Empty sequence`.

Пример входных данных #1 4 1 2 3 4 0 -2 -6 -12	Пример выходных данных #1 DUP DUP MUL SUB
Пример входных данных #2 3 1 2 3 1 11 1998	Пример выходных данных #2 Impossible
Пример входных данных #3 1 1998 1998	Пример выходных данных #3 Empty sequeunce

ЗАДАЧА F. ФУТБОЛ И ОЛИГАРХИ (TIME LIMIT 2s, MEMORY LIMIT 64M)

В последнее время некоторые российские олигархи решили поощрять футболистов раздачей мерседесов за хорошую игру. К финальной части чемпионата Европы один очень известный олигарх пообещал подарить сборной России по одному мерседесу за каждый забитый на Евро гол. Обязательным условием подарка являлся тот факт, что тренеру сборной Гусу Хиддинку должно достаться M автомобилей независимо от результата.

Как ни странно, но олигарх сдержал обещание - после окончания Евро на одной из стоянок стоял ряд новых машин ровно по количеству забитых голов.

Каждый из N игроков сборной России решил независимо от другого посетить стоянку и забрать свою долю выигрыша. Первый игрок прибыл на стоянку, разделил все количество автомобилей на N (при этом осталось ровно M автомобилей) и забрал себе свою долю. M автомобилей он благополучно оттранспортировал Хиддинку. На следующий день за своей долей на стоянку прибыл другой игрок, ничего не зная о разделе первого. Он проделал такую же манипуляцию с автомобилями. Так сделали все игроки сборной России. Каждый из них взял себе $1/N$ автомобилей из того набора, который он нашел на стоянке, и каждый отдал M автомобилей Хиддинку.

В условленный день все игроки прибыли на стоянку и поделили поровну оставшиеся автомобили, снова отдав M автомобилей Хиддинку. Естественно, никто не признался, что уже посещал стоянку и брал свою долю. При этом всякий раз игрокам доставался хотя бы один «Мерседес»,

Необходимо восстановить справедливость и выяснить, какое наименьшее возможное количество автомобилей подарил олигарх команде,

ВХОДНЫЕ ДАННЫЕ

В единственной строке входного файла содержатся два числа, разделенных одним пробелом: N – число игроков сборной ($1 < N \leq 10$), и M - количество автомобилей, которое доставалось Хиддинку при каждом дележе ($1 \leq M \leq 100$).

ВЫХОДНЫЕ ДАННЫЕ

Выходной файл должен содержать единственное число – начальное количество автомобилей на стоянке.

Пример входных данных 3 1	Пример выходных данных 79
------------------------------	------------------------------