

Задача А. Компьютерные игры

Имя входного файла: `games.in`
Имя выходного файла: `games.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт
Максимальная оценка: 100 баллов

Вася — начинающий хакер. Недавно он написал программу, которая сохраняет журнал всех нажатий клавиш на компьютере. Мечтая поскорее опробовать ее в деле, он запустил ее перед уходом в школу, чтобы узнать все пароли и секретные данные своего младшего брата Пети.

Придя из школы, Вася обнаружил, что, судя по журналу, Петя действительно использовал компьютер, но вряд ли он вводил какие-либо пароли. Скорее всего, Петя просто запустил какую-то игру, и немного в нее поиграл. Однако, программа Васи не записывает в журнал щелчки мыши, поэтому Вася не может с ходу определить, что это была за игра. Вася решил написать программу, которая бы это выяснила.

Васе известно, какими клавишами ведется управление в каждой из игр. Однако, его задание усложняется тем, что Петя мог иногда промахиваться мимо нужных клавиш управления. Вася решил принять за рабочую гипотезу утверждение, что Петя промахнулся не более чем в 10% нажатий клавиш.

Вашей задачей будет повторить достижение Васи, написав программу, которая определяет, во что мог играть Петя.

Формат входного файла

В первой строке входного файла задано N — количество игр, установленных на компьютере ($1 \leq N \leq 10\,000$). Далее следуют N строк, в которых записаны название каждой из игр и клавиши, используемые в игре для управления. Название — это непустая строка из не более 20 больших и маленьких латинских букв. Описание клавиш — это непустая строка, отделенная от названия одним пробелом, и состоящая не более чем из 26 различных маленьких латинских букв. Названия игр во входном файле различны.

В последней строке входного файла содержится журнал нажатий клавиш — строка, состоящая из L маленьких латинских букв ($1 \leq L \leq 100\,000$).

Формат выходного файла

Выведите в выходной файл названия всех игр, в которые мог играть Петя, если известно, что он промахнулся не более чем в 10% нажатий. Названия нужно выводить в том порядке, в котором они даны во входном файле. Если Петя не мог играть ни в какую из игр, выведите `NO SOLUTION`.

Примеры

<code>games.in</code>	<code>games.out</code>
4 Doom awsd Warcraft ahpmsbf Civilization rmic Quake awsdzc adwdqdsaawwzdsaaawws	Doom Quake
1 Doom awsd programmamacrobeginand	NO SOLUTION

Примечание

Решение, которое корректно работает при ограничениях $N \leq 100$, $L \leq 100$, будет получать не менее 70 баллов. Решение, которое корректно работает только на тестах с ответом `NO SOLUTION`, будет получать 0 баллов.

Задача В. Бильярд

Имя входного файла:	pool.in
Имя выходного файла:	pool.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт
Максимальная оценка:	100 баллов

Одна крупная компания разрабатывает робота для игры в бильярд. Вашей группе поручено важное задание в рамках этого проекта — написать программу, которая будет определять результат пущенного роботом шара.

Бильярд представляет собой прямоугольный стол $m \times n$ сантиметров, на котором расположены K шаров радиуса R сантиметров каждый. Вашей программе будет передано расположение этих шаров на столе, номер катящегося шара A и его направление движения (dx, dy) , а также число L , обозначающее максимальное расстояние в сантиметрах, которое может прокатиться шар.

Вашей программе требуется определить, какое из трех событий произойдет раньше: шар остановится, шар коснется другого шара или шар коснется бортика.

Формат входного файла

В первой строке входного файла заданы шесть целых чисел m, n, K, R, L, A ($5 \leq m \leq 1000$, $5 \leq n \leq 1000$, $1 \leq K \leq 100$, $1 \leq R \leq 20$, $1 \leq L \leq 1000$, $1 \leq A \leq K$). Во второй строке задаются два целых числа dx и dy , по модулю не превосходящие 100, обозначающие направление, в котором движется шар. Эти два числа обозначают, на сколько бы сдвинулся шар через некоторое время, если бы на плоскости не было никаких препятствий. Хотя бы одно из этих чисел отлично от нуля. Последние K строк файла содержат целые координаты шаров x_i и y_i ($0 \leq x_i \leq m$, $0 \leq y_i \leq n$).

Гарантируется, что шары в начальном положении не касаются друг друга и не касаются бортика.

Формат выходного файла

В выходной файл выведите одно из следующих трех слов, соответствующих событию, которое произошло раньше:

- BOARD, если шар врезался в бортик.
- TOUCH, если шар коснулся другого шара.
- STOP, если шар остановился (прокатился L сантиметров без препятствий).

Гарантируется, что одновременно с первым событием не произойдет никакого другого.

Примеры

pool.in	pool.out
15 15 2 1 10 1 1 1 2 2 11 11	STOP
15 15 2 1 11 1 1 1 2 2 11 11	TOUCH
15 15 2 1 10 1 1 -1 2 2 11 11	BOARD
15 15 2 1 10 1 1 0 2 2 6 4	TOUCH

Примечание

Решение, выводящее всегда одинаковый ответ, будет получать 0 баллов.

Задача С. Баянический квадрат

Имя входного файла: `square.in`
Имя выходного файла: `square.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт
Максимальная оценка: 100 баллов

Будем рассматривать квадраты размера $(n + 1) \times (n + 1)$, клетки которых нумеруются от $(0, 0)$ до (n, n) (на рисунке изображен такой квадрат при $n = 2$):

	0	1	2
0	(0, 0)	(1, 0)	(2, 0)
1	(0, 1)	(1, 1)	(2, 1)
2	(0, 2)	(1, 2)	(2, 2)

В клетках этого квадрата расставляются вещественные числа от -10 до 10 . Квадрат с расставленными в нем числами назовем *баяническим*, если для каждой его внутренней (то есть имеющей ровно четырех соседей) клетки верно, что значение в ней равно среднему арифметическому значений этих соседей, т.е. справедлива формула $C_{xy} = (C_{x-1,y} + C_{x+1,y} + C_{x,y+1} + C_{x,y-1})/4$. К примеру, квадрат

1	2	5	6
2	3	4	5
5	4	3	2
6	5	2	1

является баяническим. Напишите программу, которая бы по известным числам в граничных клетках баянического квадрата строила бы сам квадрат (гарантируется, что решение существует и единственно).

Формат входного файла

В первой строке входного файла задано число n ($2 \leq n \leq 50$). В последующих строках идут значения чисел на границе квадрата в порядке обхода по часовой стрелке, начиная с клетки $(0, 0)$. Например, для $n = 2$ числа будут идти в таком порядке: $(0, 0)$ $(1, 0)$ $(2, 0)$ $(2, 1)$ $(2, 2)$ $(1, 2)$ $(0, 2)$ $(0, 1)$. Все числа разделены произвольным количеством пробелов и/или переводов строки.

Формат выходного файла

В выходном файле должен содержаться искомый баянический квадрат, по $n + 1$ числу на строку. Каждое число необходимо выводить с пятью десятичными знаками после запятой. Числа в выходном файле расположены в том же порядке, что и в квадратах на рисунке. Строки расположены в порядке возрастания значения y . Соседние числа в строке отделяются друг от друга одним пробелом.

Пример

<code>square.in</code>	<code>square.out</code>
3	1.00000 2.00000 5.00000 6.00000
1 2 5 6	2.00000 3.00000 4.00000 5.00000
5 2	5.00000 4.00000 3.00000 2.00000
1 2 5 6 5 2	6.00000 5.00000 2.00000 1.00000