

Задача А. Гонки на машинках

Имя входного файла: `cars.in`
Имя выходного файла: `cars.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт
Описание тестов: 25 тестов по 4 балла

Как и у каждого мальчика, у Феде есть игрушечные машинки. Однако ему повезло больше, чем обычному мальчику — все n его машинок являются радиоуправляемыми. Целыми днями он может устраивать различные автогонки и играть с друзьями.

Из всех видов гонок Федя предпочитает гонки по прямой. В данном формате соревнования трасса имеет форму прямой и является бесконечной (соревнования идут до тех пор, пока Феде это не надоест). Изначально каждая из n машинок находится на некотором расстоянии от старта — имеет *фору* x_i метров. По команде все машинки начинают свое движение от старта, при этом каждая машинка движется во время гонки с постоянной скоростью v_i метров в секунду. Все машинки движутся в одном направлении — удаляются от старта.

Недавно Феде подарили видеокамеру, и он хочет заснять яркие моменты гонки. Прежде всего Федя хочет запечатлеть первый обгон гонки, то есть первый момент времени, в который две машины находятся на одном расстоянии от старта.

Так как этого события можно ждать очень долго, Федя хочет настроить камеру на автоматическое включение во время обгона. Однако, Федя самостоятельно не может найти время, которое пройдет со времени начала гонки до времени первого обгона. Помогите Феде — напишите программу, находящую искомую величину.

Формат входного файла

В первой строке входного файла содержится единственное число n — количество машинок на трассе ($2 \leq n \leq 100$). Каждая из следующих n строк содержит по два целых числа x_i и v_i — расстояние от старта (в метрах) и скорость машинки i (в метрах в секунду) соответственно ($1 \leq x_i, v_i \leq 1000$).

Исходно никакие две машинки не находятся в одной точке. Гарантируется, что хотя бы один обгон во время гонки произойдет.

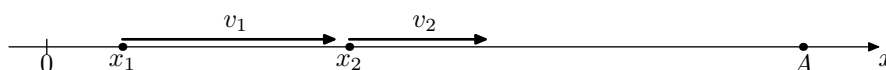
Формат выходного файла

В выходной файл выведите количество секунд, которое пройдет с момента старта до момента первого обгона, с точностью не менее 5 знаков после десятичной точки.

Примеры

<code>cars.in</code>	<code>cars.out</code>
2 1 3 4 2	3.00000
2 12 20 2 21	10.00000

Пояснение для первого примера:



На рисунке точкой A обозначено место обгона.

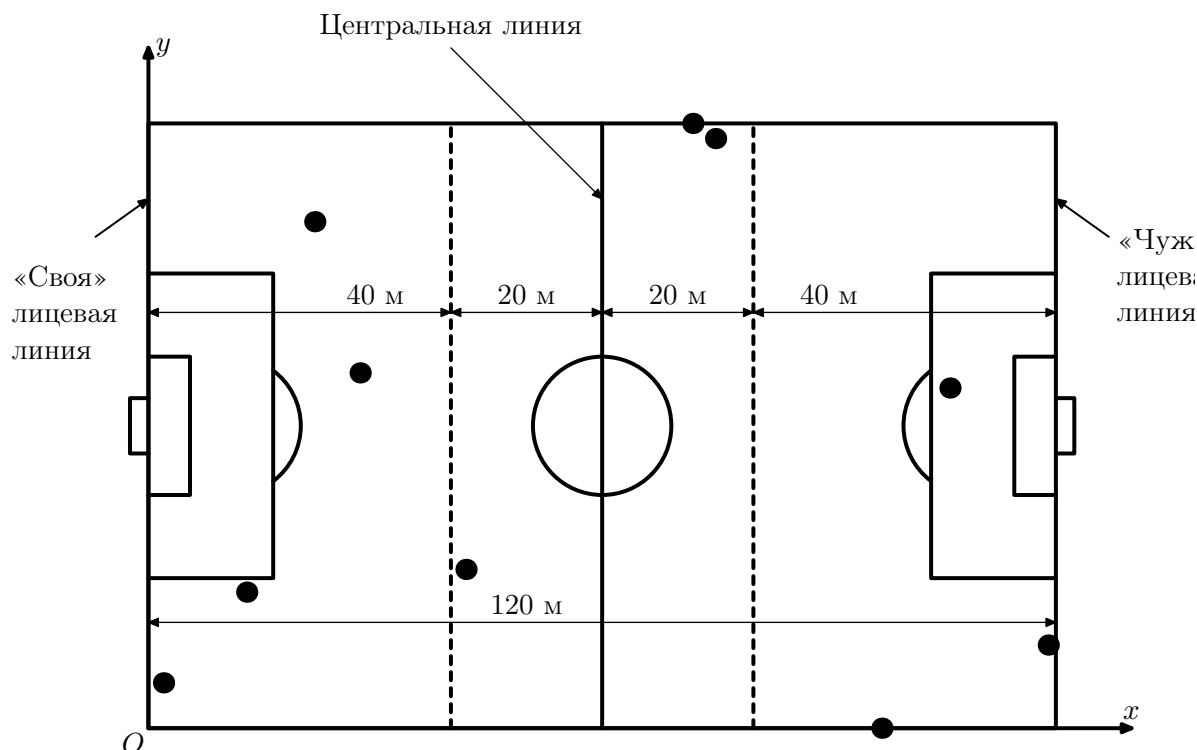
Задача В. Схема игры

Имя входного файла:	formation.in
Имя выходного файла:	formation.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт
Описание тестов:	10 тестов по 10 баллов

В тактике футбола одним из основных понятий является *схема игры*. Она определяет, сколько из десяти полевых игроков будут играть в защите, сколько — в полузащите и сколько — в нападении. Например, схема игры 5-3-2 означает, что в команде пять защитников, три полузащитника и два нападающих. В соответствии с современными представлениями на схему игры накладываются следующие ограничения: должно быть не менее одного и не более пяти защитников, не менее одного и не более пяти полузащитников и не более трех нападающих. Отметим, что нападающих может в команде и не быть совсем. Будем рассматривать только такие схемы.

Будем считать, что футбольное поле имеет длину 120 метров и ширину 80 метров. Введем на нем прямоугольную декартову систему координат таким образом, как показано на рисунке. Ворота рассматриваемой нами команды находятся слева.

Будем также считать, что игрок в некоторый момент времени находится в линии полузащиты, если он находится на расстоянии не более 20 метров от центральной линии. Соответственно, игрок находится в линии защиты, если он находится не более чем в 40 метрах от «своей» лицевой линии, и в линии нападения, если находится не более чем в 40 метрах от «чужой» лицевой линии.



Например, в ситуации, изображенной на рисунке, в линии защиты находятся четыре игрока, в линии полузащиты — три, в линии нападения — также три.

В процессе игры некоторые игроки могут перемещаться из одной линии в другую. В этой задаче будем считать, что возможно перемещение из полузащиты в защиту (и обратно) и из полузащиты в нападение (и обратно). Таким образом, игрок, который в соответствии со схемой игры является защитником, не может оказаться в линии нападения, и наоборот — игрок, который в соответствии со схемой игры является нападающим, не может оказаться в линии защиты. Кроме этого, в соответствии с установкой тренера из каждой линии в каждую могло перейти не более двух игроков.

Ваша задача состоит в том, чтобы написать программу, которая по положениям игроков в некоторый момент времени найдет все возможные схемы игры, при которых в течение игры могло возникнуть такое расположение игроков.

Формат входного файла

Входной файл содержит десять строк, содержащих по два целых числа x_i и y_i каждая, — координаты каждого из игроков команды ($0 \leq x_i \leq 120$, $x_i \neq 40$, $x_i \neq 80$, $0 \leq y_i \leq 80$).

Формат выходного файла

В первой строке выходного файла выведите k — число схем игры, по которым может играть команда. В последующих k строках в произвольном порядке выведите описание каждой из этих схем. Следуйте формату данных, приведенному в примере.

Примеры

formation.in	formation.out
97 0	9
13 18	2-5-3
2 6	3-5-2
119 11	3-4-3
42 21	4-5-1
72 80	4-4-2
75 78	4-3-3
106 45	5-4-1
22 67	5-3-2
28 47	5-2-3

Пример соответствует приведенному выше рисунку.

Задача С. Таблица

Имя входного файла:	table.in
Имя выходного файла:	table.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт
Описание тестов:	25 тестов по 4 балла

В новой математической игре для одного игрока «Таблица чисел» используется таблица размером n строк на m столбцов, заполненная целыми числами. За один ход игрок выбирает одну из строк или один из столбцов и меняет знак на противоположный у всех чисел этой строки или столбца.

Цель игры состоит в том, чтобы привести таблицу в такой вид, что сумма чисел в каждой строке и каждом столбце является неотрицательной. При этом минимизировать количество ходов не требуется, но можно сделать не более 20000 ходов.

Ваша задача состоит в том, чтобы написать программу, которая по описанию начального вида таблицы, найдет последовательность ходов, которая ведет к достижению цели игры, или определит, что цель игры недостижима.

Формат входного файла

Первая строка входного файла содержит два целых числа: n и m ($2 \leq n, m \leq 100$). Каждая из последующих n строк содержит по m целых чисел $a_{i,j}$ ($|a_{i,j}| \leq 100$).

Формат выходного файла

В первой строке выходного файла выведите k — число ходов, которые необходимо выполнить, или «-1», если цели игры добиться невозможно. В первом случае в последующих k строках выведите описание этих ходов.

Описание каждого хода должно быть выведено в отдельной строке, которые должны содержать: тип хода («R» — на этом ходе выбирается строка или «C» — на этом ходе выбирается столбец) и номер соответствующей строки или столбца. Строки и столбцы нумеруются натуральными числами начиная с единицы, строки нумеруются сверху вниз, а столбцы — слева направо.

Выведенная вашей программой последовательность ходов должна содержать не более 20000 ходов. Гарантируется, что если существует последовательность ходов, ведущая к достижению цели игры, то существует и последовательность, удовлетворяющая указанному ограничению.

Примеры

table.in	table.out
2 2 -1 -3 1 -2	1 C 2
3 2 -1 -1 -1 -1 -1 -1	3 R 1 R 2 R 3

Задача D. Кодовый замок

Имя входного файла: `code.in`
Имя выходного файла: `code.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт
Описание тестов: 50 тестов по 2 балла

Компания «Замки и замки» недавно разработала новый тип кодового замка, для размещения на воротах замков. Панель замка представляет собой прямоугольник шириной w ячеек и высотой h ячеек. В некоторых из них расположены кнопки.

Код на этом замке вводится одновременным нажатием k кнопок. Для того, чтобы код было легче запомнить, используемые в нем кнопки должны образовывать связную область. Область называется *связной*, если из любой клетки области можно добраться до любой другой, перемещаясь только между клетками этой области с общей стороной. Важным критерием надежности замка является число различных кодов, которые на нем можно набрать.

Для оценки надежности замков требуется написать программу для вычисления указанной величины.

Формат входного файла

В первой строке входного файла находятся три целых числа h , w и k ($1 \leq h, w \leq 30$; $1 \leq k \leq 10$). Каждая из последующих h строк содержит w символов. Символ «#» обозначает кнопку, а «.» — ее отсутствие.

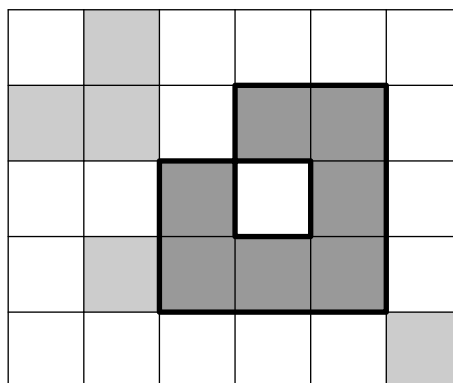
Формат выходного файла

В выходной файл выведите единственное число — количество кодов, удовлетворяющих указанным требованиям.

Примеры

<code>code.in</code>	<code>code.out</code>
2 2 2 #. ##	2
5 6 7 .#.... ##.##. ..#.#. .####.#	3

На рисунке изображен один из возможных кодов для второго примера.



Индивидуальная олимпиада по информатике и программированию
Очный тур, 28 марта 2010 года

В таблице приведены значения чисел h , w и k для всех тестов, на которых будет выполняться проверка вашего решения.

Номер теста	h	w	k	Номер теста	h	w	k
1	$h = 1$	$w = 1$	$k = 1$	26	$h = 6$	$w = 9$	$k = 7$
2	$h = 30$	$w = 29$	$k = 1$	27	$h = 3$	$w = 24$	$k = 7$
3	$h = 5$	$w = 7$	$k = 2$	28	$h = 24$	$w = 1$	$k = 7$
4	$h = 12$	$w = 14$	$k = 2$	29	$h = 10$	$w = 20$	$k = 7$
5	$h = 27$	$w = 30$	$k = 2$	30	$h = 26$	$w = 7$	$k = 7$
6	$h = 3$	$w = 10$	$k = 3$	31	$h = 2$	$w = 8$	$k = 8$
7	$h = 6$	$w = 5$	$k = 3$	32	$h = 28$	$w = 18$	$k = 8$
8	$h = 7$	$w = 3$	$k = 3$	33	$h = 5$	$w = 5$	$k = 8$
9	$h = 24$	$w = 27$	$k = 3$	34	$h = 27$	$w = 19$	$k = 8$
10	$h = 16$	$w = 9$	$k = 3$	35	$h = 25$	$w = 17$	$k = 8$
11	$h = 8$	$w = 9$	$k = 4$	36	$h = 10$	$w = 10$	$k = 9$
12	$h = 24$	$w = 14$	$k = 4$	37	$h = 17$	$w = 27$	$k = 9$
13	$h = 30$	$w = 16$	$k = 4$	38	$h = 13$	$w = 11$	$k = 9$
14	$h = 7$	$w = 9$	$k = 4$	39	$h = 7$	$w = 16$	$k = 9$
15	$h = 30$	$w = 4$	$k = 4$	40	$h = 30$	$w = 30$	$k = 9$
16	$h = 3$	$w = 8$	$k = 5$	41	$h = 30$	$w = 30$	$k = 10$
17	$h = 24$	$w = 20$	$k = 5$	42	$h = 30$	$w = 30$	$k = 10$
18	$h = 15$	$w = 19$	$k = 5$	43	$h = 29$	$w = 30$	$k = 10$
19	$h = 10$	$w = 10$	$k = 5$	44	$h = 30$	$w = 30$	$k = 10$
20	$h = 30$	$w = 30$	$k = 5$	45	$h = 30$	$w = 30$	$k = 10$
21	$h = 6$	$w = 4$	$k = 6$	46	$h = 30$	$w = 30$	$k = 10$
22	$h = 28$	$w = 12$	$k = 6$	47	$h = 30$	$w = 30$	$k = 10$
23	$h = 6$	$w = 17$	$k = 6$	48	$h = 30$	$w = 30$	$k = 10$
24	$h = 8$	$w = 8$	$k = 6$	49	$h = 30$	$w = 30$	$k = 10$
25	$h = 14$	$w = 13$	$k = 6$	50	$h = 30$	$w = 30$	$k = 10$