Задача A. Робот

Имя входного файла:	`robot.in`
Имя выходного файла:	`robot.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Роботу дали задание пройти по лабиринту. Этот лабиринт представляет из себя прямоугольное поле размера $m \times n$ клеток, в котором каждая клетка либо занята сплошной стеной, либо свободна. Каждая клетка описывается парой координат (x, y), где $1 \leqslant x \leqslant m$ и $1 \leqslant y \leqslant n$. Особенности конструкции робота таковы, что он из клетки (x, y) может попасть только в клетки $(x-1, y)$, $(x+1, y)$ и $(x, y+1)$, конечно, если они свободны, но не может уменьшить свою координату по y и перейти на клетку $(x, y-1)$.

Робот хочет провести в лабиринте как можно больше времени, так как после прохождения лабиринта его наверняка заставят заниматься чем-то более тяжёлым. Однако, если он остановится хоть на секунду или просто замедлит свое движение, могут возникнуть подозрения, что он сломался, и тогда ему грозит попадание на свалку металлолома. Поэтому робот хочет выбрать путь, который бы имел наибольшую длину, и пройти по нему с постоянной скоростью.

Задача робота усложняется тем, что ему нельзя возвращаться на клетку, в которой он уже побывал — иначе баг в программе, которая записывает его путь, заставит его крутиться на одном месте, а о последствиях такой ошибки страшно даже подумать!

Помогите роботу найти самый длинный путь по лабиринту, не имеющий самопересечений и такой, что в нём не уменьшается координата y. Путь может начинаться в любой свободной клетке с $y = 1$ и заканчиваться в любой свободной клетке с $y = n$; можно считать, что весь лабиринт огорожен сплошной стеной, и выходить за его пределы нельзя.

Формат входного файла

В первой строке входного файла заданы целые числа m и n через пробел ($1 \leqslant m, n \leqslant 100$). В последующих n строках содержится по m символов в каждой; j-ый символ i-ой из этих строк равен '`X`' (икс большое), если соответствующая клетка занята стеной, и '`.`' (точка), если она свободна.

Формат выходного файла

Если пути с указаными свойствами не существует, выведите в выходной файл число `-1`. В противном случае в первую строку выведите число посещённых роботом клеток k, а в последующие k строк по паре целых чисел $x_i\, y_i$ через пробел — координаты клеток пути в порядке их посещения. Число k должно быть максимально; если оптимальных ответов несколько, разрешается вывести любой из них.

Примеры

robot.in	robot.out

`2 2`
`..`
`..`
	`4`
	`1 1`
	`2 1`
	`2 2`
	`1 2`

`3 4`
`.X.`
`...`
`X.X`
`..X`
	`6`
	`3 1`
	`3 2`
	`2 2`
	`2 3`
	`2 4`
	`1 4`

`1 1` `X`	`-1`

Задача B. Разведчик

Имя входного файла:	`scout.in`
Имя выходного файла:	`scout.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Разведчик Василий проник в штаб врага, и ему удалось увидеть план, на котором было указано расположение секретных вражеских сил. К сожалению, при уходе от погони Василий сильно стукнулся головой, и теперь он не может вспомнить, где же именно находятся силы врага. Всё, что он запомнил — это то, что положение крестика на карте, обозначавшего расположение вражеских сил, можно было получить, если пересечь два отрезка AB и CD, где A, B, C и D — различные крупные населённые пункты.

Враг готовится к наступлению, и необходимо во что бы то ни стало определить местонахождение вражеских секретных сил и нанести упреждающий удар. Первое, что постановили сделать на совещании — определить, сколько же всего таких точек, являющихся пересечениями двух отрезков между различными крупными населёнными пунктами, можно поставить на карту. Населённые пункты считаются точками независимо от их величины.

Найдите число таких точек, и в зависимости от ответа командование решит, что делать дальше.

Формат входного файла

В первой строке задано целое число N $(0 \leqslant N \leqslant 60)$. В последующих N строках содержится по два целых числа $X_k\,Y_k$ через пробел — координаты k-ого крупного населённого пункта на карте $|X_k|, |Y_k| \leqslant 1000$. Известно, что в этой местности никакие три крупных населённых пункта не лежат на одной прямой.

Формат выходного файла

Выведите в выходной файл одно число M — количество различных точек на карте, являющихся пересечениями двух отрезков между четырьмя различными крупными населёнными пунктами.

Примеры

scout.in	scout.out
4 0 1 1 0 2 1 1 2	1
1 5 5	0
5 0 5 4 10 10 8 10 2 4 0	5

Задача C. Делители

Имя входного файла:	`divisors.in`
Имя выходного файла:	`divisors.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

По данному числу N определите количество его различных положительных делителей.

Формат входного файла

В первой строке входного файла задано единственное целое число N ($1 \leqslant N \leqslant 10^{15}$).

Формат выходного файла

Выведите единственное число k — количество различных положительных делителей числа N.

Примеры

divisors.in	divisors.out
1	1
2	2
6	4
29	2
48	10

В последнем примере число 48 имеет десять делителей — это числа 1, 2, 3, 4, 6, 8, 12, 16, 24 и 48.