

Задача А. Игра с шариками

Имя входного файла: `balls.in`
Имя выходного файла: `balls.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Действие одной очень популярной компьютерной игры, которая часто бывает установлена на карманных компьютерах, происходит на квадратном поле размером 11 на 11, разбитом на 121 маленький квадратик.

Изначально в каждом квадратике находится шарик одного из пяти цветов: красного (обозначается символом **R**), синего (**B**), зеленого (**G**), желтого (**Y**), фиолетового (**V**). Назовем *связной областью*, содержащей данный шарик, все шарики, до которых можно добраться из данного, двигаясь каждый раз на один квадратик по вертикали или горизонтали, не выходя за границы игрового поля и проходя только по шарикам того же цвета, что и данный.

При выборе некоторого шарика автоматически выбираются все шарики, лежащие в одной связной области с ним. Если эта связная область содержит хотя бы 2 шарика, то эти шарики исчезают и игроку начисляется $n \cdot (n - 1)$ очков, где n — количество шариков в связной области.

Задано начальное расположение шариков. Необходимо для каждого цвета определить, какое максимальное количество очков можно набрать за первый ход, выбрав один шарик такого цвета.

Формат входных данных

Входной файл содержит 11 строк по 11 символов в каждой — описание игрового поля.

Формат выходных данных

Для каждого цвета шариков в выходной файл выведите максимальное количество очков, которое можно набрать, выбрав шарик этого цвета. Следуйте формату, приведенному в примере.

Примеры

balls.in	balls.out
RRRRRBBBGGG RRRRRBBBGGG RRRRRBBBGGG RRRRRBBBGGG RRRRRBBBGGG RRRRRBBBGGG RRRRRBBBGGG YYYYYYYYYYY YYYYYYYYYYY YYYYYYYYYYY VRVRVBVBVGV	R: 1190 G: 420 B: 420 Y: 1056 V: 0

Задача В. Ферзь в угол

Имя входного файла: `queen1.in`
Имя выходного файла: `queen1.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

В левом нижнем углу доски $M \times N$ стоит ферзь. Двое игроков по очереди ходят ферзем, перемещая его на любое число клеток по вертикали вверх, по горизонтали вправо, или по диагонали вправо-вверх. Выигрывает тот, кто поставит ферзя в правый верхний угол доски. Определите, какой из игроков имеет выигрышную стратегию.

Формат входных данных

На вход программе подается два натуральных числа M и N , не превосходящих 100.

Формат выходных данных

Программа должна вывести номер игрока (1 или 2), который имеет выигрышную стратегию.

Примеры

queen1.in	queen1.out
3 4	1

Задача С. Игрушечный лабиринт

Имя входного файла: `labirint.in`
Имя выходного файла: `labirint.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Игрушечный лабиринт представляет собой прозрачную плоскую прямоугольную коробку, внутри которой есть препятствия и перемещается шарик. Коробку можно наклонять влево, вправо, к себе или от себя, после каждого наклона шарик перемещается в заданном направлении до ближайшего препятствия или до стенки лабиринта, после чего останавливается. Целью игры является загнать шарик в одно из специальных отверстий-выходов. Шарик проваливается в отверстие, если оно встречается на его пути (шарик не обязан останавливаться в отверстии).

Первоначально шарик находится в левом верхнем углу лабиринта. Гарантируется, что решение существует и левый верхний угол не занят препятствием или отверстием.

Формат входных данных

В первой строке входного файла записаны числа N и M — размеры лабиринта (целые положительные числа, не превышающие 100). Затем идет N строк по M чисел в каждой — описание лабиринта. Число 0 в описании означает свободное место, число 1 — препятствие, число 2 — отверстие.

Формат выходных данных

Выведите единственное число — минимальное количество наклонов, которые необходимо сделать, чтобы шарик покинул лабиринт через одно из отверстий.

Примеры

labirint.in	labirint.out
4 5 0 0 0 0 1 0 1 1 0 2 0 2 1 0 0 0 0 1 0 0	3

Задача D. Кони ходят по очереди

Имя входного файла: `knight3.in`
Имя выходного файла: `knight3.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Переведите каждого из двух коней из одной клетки в другую за наименьшее общее число ходов. Два коня не могут одновременно находиться в одной клетке. **Ходы коней должны чередоваться.**

Формат входных данных

Во входном файле записаны координаты первого и второго коня, затем координаты клеток, куда нужно их переместить.

Формат выходных данных

Программа должна вывести последовательность ходов коней в виде нескольких строк. Первым символом в строке должен быть номер коня (1 или 2), затем, через пробел, координаты клетки, в которую он переставляется. Необходимо вывести любое из возможных оптимальных решений. Кони должны ходить по очереди, первым может ходить любой из коней, кони могут сделать различное число ходов.

Примеры

knight3.in	knight3.out
a1	1 b3
c2	2 a1
c2	1 d4
a1	2 b3
	1 c2
	2 a1

Задача E. Только направо

Имя входного файла: `nolefts.in`
Имя выходного файла: `nolefts.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Змей Горыныч оказался в лабиринте и хочет выбраться из него как можно скорее. К сожалению, после вчерашнего злоупотребления кефиром левая голова Змея соображает

плохо. Поэтому Змей Горыныч может поворачивать направо и идти прямо, но не может поворачивать налево и разворачиваться на месте. Помогите Змею Горынычу определить длину кратчайшего пути до выхода из лабиринта.

Формат входных данных

В первой строке через пробел записаны числа r и c ($4 \leq r, c \leq 20$) — количество строк и столбцов в карте лабиринта. В каждой из следующих r строк записано по c символов, задающих эту карту. Символ `S` обозначает положение Змея Горыныча, символ `F` — точку выхода из лабиринта, символ `X` — стенку. Пробелами обозначены проходимые клетки. Гарантируется, что лабиринт окружен стенами. Перед началом движения Змей Горыныч может сориентироваться по любому из 4 направлений (вверх, вниз, влево или направо).

Формат выходных данных

Выведите единственное число — расстояние, которое придется пройти Змею Горынычу. Гарантируется, что он всегда сможет выйти из лабиринта.

Примеры

nolefts.in	nolefts.out
10 14 XXXXXXXXXXXXX X XXX X XFXXXXX X XXX XX XX X X S X XX XXXXXX X X X X X X X X X X X XXX XX X XXXXXXXXXXXXX	29