Задача A. 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 B. Игрушечный лабиринт

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

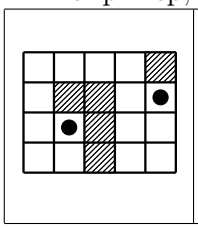
Игрушечный лабиринт представляет собой прозрачную плоскую прямоугольную коробку, внутри которой есть препятствия и перемещается шарик. Коробку можно наклонять влево, вправо, к себе или от себя, после каждого наклона шарик перемещается в заданном направлении до ближайшего препятствия или до стенки лабиринта, 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.

Например, лабиринту, изображенному на рисунке, будет соответствовать такое описание:



```
4 5
0 0 0 0 1
0 1 1 0 2
0 2 1 0 0
0 0 1 0 0
```

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 C. Числа

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

Витя хочет придумать новую игру с числами. В этой игре от игроков требуется преобразовывать четырехзначные числа не содержащие нулей при помощи следующего разрешенного набора действий:

1. Можно увеличить первую цифру числа на 1, если она не равна 9.
2. Можно уменьшить последнюю цифру на 1, если она не равна 1.
3. Можно циклически сдвинуть все цифры на одну вправо.
4. Можно циклически сдвинуть все цифры на одну влево.

Например, применяя эти правила к числу 1234, можно получить числа 2234, 1233, 4123 и 2341 соответственно.

Точные правила игры Витя пока не придумал, но пока его интересует вопрос, как получить из одного числа другое за минимальное количество операций.

Формат входных данных

Во входном файле содержится два различных четырехзначных числа, каждое из которых не содержит нулей.

Формат выходных данных

Программа должна вывести последовательность четырехзначных чисел, не содержащих нулей. Последовательность должна начинаться первым из данных чисел и заканчиваться вторым из данных чисел, каждое последующее число в последовательности должно быть получено из предыдущего числа применением одного из правил. Количество чисел в последовательности должно быть минимально возможным.

Примеры

numbers.in	numbers.out
9876 8876	9876 8769 8768 8876

Задача D. 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 ($4 \leqslant r, c \leqslant 20$) — 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 XXXXXXXXXXXXXX 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 XXXXXXXXXXXXXX	29

Задача E. Цивилизация

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

Карта мира в компьютерной игре «Цивилизация» версии 1 представляет собой прямоугольник, разбитый на квадратики. Каждый квадратик может иметь один из нескольких возможных рельефов, для простоты ограничимся тремя видами рельефов — поле, лес и вода. Поселенец перемещается по карте, при этом на перемещение в клетку, занятую полем, необходима одна единица времени, на перемещение в лес — две единицы времени, а перемещаться в клетку с водой нельзя.

У вас есть один поселенец, вы определили место, где нужно построить город, чтобы как можно скорее завладеть всем миром. Найдите маршрут переселенца, по которому можно прийти в место строительства города за минимальное время. На каждом ходе переселенец может перемещаться в клетку, имеющую общую сторону с той клеткой, где он сейчас находится.

Формат входных данных

Во входном файле записаны два натуральных числа N и M, не превосходящих 1000 — размеры карты мира (N — число строк в карте, M — число столбцов). Затем заданы координаты начального положения поселенца x и y, где x — номер строки, y — номер столбца на карте ($1 \leqslant x \leqslant N$, $1 \leqslant y \leqslant M$), строки нумеруются сверху вниз, столбцы — слева направо. Затем аналогично задаются координаты клетки, куда необходимо привести поселенца.

Далее идет описание карты мира в виде N строк, каждая из которых содержит M символов. Каждый символ может быть либо «`.`» (точка), обозначающим поле, либо «`W`», обозначающим лес, либо «`#`», обозначающим воду.

Гарантируется, что начальная и конечная клетки пути переселенца не являются водой.

Формат выходных данных

В первой строке выходного файла выведите количество единиц времени, необходимое для перемещения поселенца (перемещение в клетку с полем занимает 1 единицу времени, перемещение в клетку с лесом — 2 единицы времени). Во второй строке выходного файла выведите последовательность символов, задающих маршрут переселенца. Каждый символ должен быть одним из четырех следующих: «`N`» (движение вверх), «`E`» (движение вправо), «`S`» (движение вниз), «`W`» (движение влево). Если таких маршрутов несколько — выведите любой из них.

Если дойти из начальной клетки в конечную невозможно, выведите число -1.

Примеры

civ.in	civ.out
4 8 1 1 4 8 ....WWWW .######. .#..W... ...WWWW.	13 SSSEENEEEEES
4 7 2 2 3 6 ####### #WW#..# #WW#..# #######	-1

Задача F. Цивилизация (версия для Python)

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

Карта мира в компьютерной игре «Цивилизация» версии 1 представляет собой прямоугольник, разбитый на квадратики. Каждый квадратик может иметь один из нескольких возможных рельефов, для простоты ограничимся тремя видами рельефов — поле, лес и вода. Поселенец перемещается по карте, при этом на перемещение в клетку, занятую полем, необходима одна единица времени, на перемещение в лес — две единицы времени, а перемещаться в клетку с водой нельзя.

У вас есть один поселенец, вы определили место, где нужно построить город, чтобы как можно скорее завладеть всем миром. Найдите маршрут переселенца, по которому можно прийти в место строительства города за минимальное время. На каждом ходе переселенец может перемещаться в клетку, имеющую общую сторону с той клеткой, где он сейчас находится.

Формат входных данных

Во входном файле записаны два натуральных числа N и M, не превосходящих 500 — размеры карты мира (N — число строк в карте, M — число столбцов). Затем заданы координаты начального положения поселенца x и y, где x — номер строки, y — номер столбца на карте ($1 \leqslant x \leqslant N$, $1 \leqslant y \leqslant M$), строки нумеруются сверху вниз, столбцы — слева направо. Затем аналогично задаются координаты клетки, куда необходимо привести поселенца.

Далее идет описание карты мира в виде N строк, каждая из которых содержит M символов. Каждый символ может быть либо «`.`» (точка), обозначающим поле, либо «`W`», обозначающим лес, либо «`#`», обозначающим воду.

Гарантируется, что начальная и конечная клетки пути переселенца не являются водой.

Формат выходных данных

В первой строке выходного файла выведите количество единиц времени, необходимое для перемещения поселенца (перемещение в клетку с полем занимает 1 единицу времени, перемещение в клетку с лесом — 2 единицы времени). Во второй строке выходного файла выведите последовательность символов, задающих маршрут переселенца. Каждый символ должен быть одним из четырех следующих: «`N`» (движение вверх), «`E`» (движение вправо), «`S`» (движение вниз), «`W`» (движение влево). Если таких маршрутов несколько — выведите любой из них.

Если дойти из начальной клетки в конечную невозможно, выведите число -1.

Примеры

civ.in	civ.out
4 8 1 1 4 8 ....WWWW .######. .#..W... ...WWWW.	13 SSSEENEEEEES
4 7 2 2 3 6 ####### #WW#..# #WW#..# #######	-1