

Задача А. Маяки

Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

Мореплаватель Кирилл построил новый корабль, и хочет испытать его в открытом море. Сейчас корабль находится на причале, и для того, чтобы добраться до открытого моря, ему предстоит преодолеть большую акваторию, полную маяков.

Введем на плоскости декартову систему координат. Акваторию можно считать прямоугольником с углами в точках $(1, 1)$ и (n, m) . Перемещение корабля по акватории происходит так: изначально корабль находится у причала, расположенного в точке $(p, 0)$, и начинает свое движение в сторону точки $(p, 1)$. Каждый раз, когда корабль достигает целочисленной точки, он должен решить, в каком направлении продолжить движение. Он может либо продолжить идти в том же направлении, либо повернуть на 90 градусов в любую сторону. Развернуться корабль не может.

Открытое море расположено над акваторией. Для того, чтобы выйти в открытое море, корабль должен из любой из точек на верхней стороне акватории (то есть, имеющих y -координату, равную m) переместиться вверх (в направлении увеличения y -координаты). Из соображений безопасности корабль не может выходить за пределы акватории иным способом, то есть до достижения верхней стороны акватории.

К сожалению, акватория не идеальна, и между некоторыми парами соседних целочисленных точек невозможно пройти, так как путь преграждает огромная скала. Всего в акватории расположено k скал.

Также попадание в открытое море усложняют маяки, а также сложные правила перемещения судов. Согласно этим правилам, в каждой целочисленной точке акватории расположен маяк одного из трех цветов: белого, красного или зеленого. Каждый раз, когда корабль находится в целочисленной точке и выбирает, в каком направлении продолжить движение, маяк, находящийся в этой точке, накладывает следующие ограничения:

- Если это маяк красного цвета, то корабль обязан повернуть в этой точке налево.
- Если это маяк зеленого цвета, то корабль обязан повернуть в этой точке направо.
- Если это маяк белого цвета, корабль может повернуть в любую из сторон, а также продолжить движение прямо.

Кирилл очень не любит формальности, поэтому он хочет добраться до открытого моря, пройдя минимально возможное количество маяков. Помогите ему найти наилучший маршрут или определите, что добраться до моря невозможно.

Формат входных данных

Первая строка входных данных содержит два целых числа n и m — ширина и длина бухты ($1 \leq n, m \leq 1000$). Каждая из следующих m строк содержит по n целых чисел — описание маяков («0» означает маяк белого цвета, «1» — зеленого цвета, «2» — красного цвета). Строки даются в порядке уменьшения y -координаты, то есть маяки первой строки граничат с открытым морем, а последней — с причалом.

Следующая строка содержит единственное целое число k — количество скал ($0 \leq k \leq 10^6$). Следующие k строк содержат описания пар маяков, между которыми находится очередная скала. Каждая строка содержит четыре целых числа x_1, y_1, x_2, y_2 — координаты соответствующих маяков ($1 \leq x_1, x_2 \leq n, 1 \leq y_1, y_2 \leq m$). Гарантируется, что маяки в описании скалы являются соседними по вертикали или по горизонтали.

Последняя строка содержит единственное целое число p ($1 \leq p \leq n$) — номер причала, от которого должен отойти корабль. Отошедший от причала корабль подходит к p -му слева из граничащих с причалами маяку снизу (имеющего x -координату, равную p).

Формат выходных данных

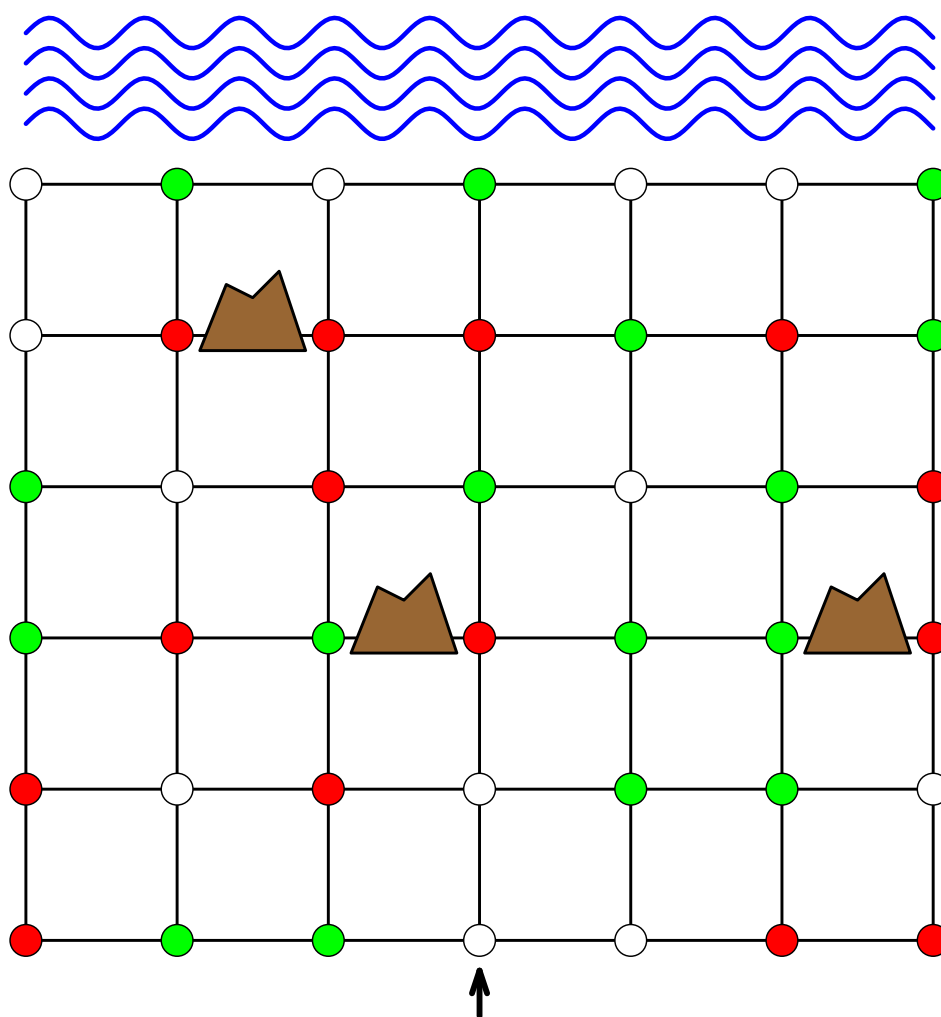
Если не существует ни одного пути, не нарушающего правил, выведите единственное число -1 . В противном случае выведите единственное число — наименьшее количество маяков, мимо которых кораблю придется проследовать на пути в открытое море.

Пример

lighthouses.in	lighthouses.out
7 6	11
0 1 0 1 0 0 1	
0 2 2 2 1 2 1	
1 0 2 1 0 1 2	
1 2 1 2 1 1 2	
2 0 2 0 1 1 0	
2 1 1 0 0 2 2	
3	
2 5 3 5	
3 3 4 3	
6 3 7 3	
4	

Замечание

Иллюстрация к тесту из примера:



Задача В. Игрушечный лабиринт

Ограничение по времени: 2 секунды
 Ограничение по памяти: 64 мегабайта

Игрушечный лабиринт представляет собой прозрачную плоскую прямоугольную коробку, внутри которой есть препятствия и перемещается шарик. Лабиринт 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.

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:

	<pre> 4 5 0 0 0 0 1 0 1 1 0 2 0 2 1 0 0 0 0 1 0 0 </pre>
--	--

Формат выходных данных

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

Пример

labirint.in	labirint.out
<pre> 4 5 0 0 0 0 1 0 1 1 0 2 0 2 1 0 0 0 0 1 0 0 </pre>	3