

Задача А. Восьминашки

Имя входного файла: `puzzle.in`
 Имя выходного файла: `puzzle.out`
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 256 мегабайт

«Восьминашки» – упрощенный вариант известной головоломки «Пятнашки». Восемь костяшек, пронумерованных от 1 до 8, расставлены по ячейкам игровой доски 3 на 3, одна ячейка при этом остается пустой. За один ход разрешается передвинуть одну из костяшек, расположенных рядом с пустой ячейкой, на свободное место. Цель игры – для заданной начальной конфигурации игровой доски за минимальное число ходов получить выигрышную конфигурацию (пустая ячейка обозначена нулем):

```

1 2 3
4 5 6
7 8 0

```

Формат входного файла

Во входном файле содержится начальная конфигурация головоломки – 3 строки по 3 числа в каждой.

Формат выходного файла

Если решение существует, то в первой строке выходного файла выведите минимальное число перемещений костяшек, которое нужно сделать, чтобы достичь выигрышной конфигурации, а во второй строке выведите соответствующую последовательность ходов: L означает, что в результате перемещения костяшки пустая ячейка сдвинулась влево, R – вправо, U – вверх, D – вниз. Если таких последовательностей несколько, то выведите любую из них. Если же выигрышная конфигурация недостижима, то выведите в выходной файл одно число –1.

Пример

<code>puzzle.in</code>	<code>puzzle.out</code>
0 1 6 4 3 2 7 5 8	8 RDRULDDR
0 1 2 3 4 5 6 7 8	22 RDLDRULLDRUURDDLURRD
1 2 3 8 0 4 7 6 5	-1

Комментарии

- Помните, что если решения не существует, алгоритм A^* обойдет ровно половину всех возможных конфигураций игрового поля, никакого ускорения по сравнению с поиском в ширину не будет. Чтобы избежать заведомо неразрешимых конфигураций, реализуйте вычисление инварианта для игрового поля, который позволяет отделить неразрешимые конфигурации от разрешимых.
- Алгоритм A^* находит кратчайшие пути только в том случае, когда используемая эвристика является *допустимой* и *монотонной*. Прежде чем применить эвристики, которые обсуждались на семинаре (число костяшек, расположенных не на своих местах; сумма манхеттенских расстояний от костяшек до их итоговых позиций), проверьте, удовлетворяют ли они необходимым условиям.
- Постарайтесь найти выигрышную конфигурацию для решения которой требуется максимальное число ходов.
- Ваш алгоритм должен уметь работать с различными эвристиками, кроме того, он должен поддерживать различные функции приоритета ($f(v) := g(v) + h(v)$ – алгоритм A^* , $f(v) := g(v)$ – поиск в ширину, $f(v) := h(v)$ – best-first search)
- Ваша реализация должна работать для произвольных размеров квадратной игровой доски, а не только 3 на 3.