

# Навигатор

Древний народ, живший в городе Мачу-Пикчу, был очень развит. У них были компьютеры, телефоны, и они умели возводить из камней невероятной сложности конструкции. Чтобы защитить свои жилища от случайных животных и менее развитых соседей, жители Мачу-Пикчу построили свой город в виде большого лабиринта. Для своей навигации по городу они хотят использовать специальные навигаторы, которые умеет строить кратчайший путь из точки А в точку Б.

Напишите программу для навигатора, поиска кратчайшего пути в городе Мачу-Пикчу для двух заданных точек. Программное обеспечение навигатора написано на языке Java и в библиотеке реализован интерфейс для поиска алгоритма:

```
/**
 * Интерфейс поиска маршрутов в городе Мачу-Пикчу
 */
public interface Navigator
{
    /**
     * Поиск кратчайшего маршрута на карте города между двумя точками
     * @param map карта города
     * @return карта города с построенным маршрутом
     */
    char[][] searchRoute(char[][] map);
}
```

Необходимо написать реализацию данного интерфейса. На вход программе подается карта города, заданная как символьный двумерный массив размером  $M \times N$  ( $1 \leq M, N \leq 10000$ ).

Обозначения на карте

- # стена
- . дорога
- @ начало маршрута
- x конец маршрута

Жители города могут перемещаться на соседние клетки по вертикали и по горизонтали. По диагонали перемещаться жители не могут. Если построить маршрут невозможно, то функция `searchRoute` должна возвращать `null`.

Результатом работы программы должен быть символьный массив с картой города и построенным маршрутом. Маршрут должен быть проложен символом '+'.

## Пример 1

### Ввод

```
...@.
.####
.....
####.
.X...
```

## Вывод

```
+++@.
####
+++++
####+
.X+++
```

## Пример 2

### Ввод

```
..X..
#####
.....
.@...
.....
```

### Вывод

```
null
```

## Пример 3

### Ввод

```
...@
#.###
.....
...X
.....
```

### Вывод

```
.+++@
#####
.+...
.+++X
.....
```

## Проверка задания

Проверка задания будет осуществляться автоматически с помощью набора тестов. Тесты буду проверять корректность ответа, а также время работы алгоритма. За каждый тест будут начисляться баллы. Помимо автотестов будет проверяться код человеком при условии, что программа работает.