

Парное программирование: Лабиринт

В данной лабораторной работе вы опробуете метод, называемый «парное программирование».

Суть метода заключается в том, что над одной программой за одним компьютером работают одновременно два программиста. Первый программист («ведомый») непосредственно пишет код, решая поставленную задачу. Второй программист («ведущий») непрерывно просматривает код, пишущийся первым, обнаруживает ошибки, вносит коррективы, продумывает архитектуру решения. Регулярно программисты меняются ролями.

Более подробная информация о методе:

<http://tech.yandex.ru/education/shri/msk-2012/talks/559/>

<http://dev.by/blogs/main/parnoe-programmirovanie-dve-golovy-luchshe>

Для выполнения задания разбейтесь на группы по два человека, распределите первоначально роли и приступайте к выполнению задания. Каждые 20 минут выполняйте смену ролей в команде.

Для сдачи лабораторной работы предъявите репозиторий на GitHub / Bitbucket / ..., содержащий ваши программные решения и модульные тесты к ним.

Языки программирования, допустимые для выполнения работы - по выбору студента: Java, C#, PHP.

Условия

Вы стоите перед входом в идеальный лабиринт. Лабиринт считается «идеальным», если удовлетворяет следующим условиям:

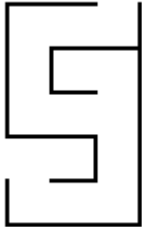
1. Он представляет собой прямоугольную сетку комнат, размером R рядов на C столбцов.
2. Лабиринт имеет две двери наружу: вход и выход. Вход всегда размещается на северной стене, выход может быть на любой стене.
3. Между любыми двумя комнатами в лабиринте существует ровно один маршрут.

Вы решили пройти алгоритм, используя алгоритм «всегда поворачивать налево», в котором при любой возможной развилке вы всегда выбираете самое левое ответвление. Если вы упираетесь в тупик, вы дважды поворачиваете направо (разворачиваетесь на 180 градусов) и продолжаете движение. (Представьте, что вы касаетесь левой рукой стены и двигаетесь по лабиринту, не отрывая руки от стены, таким образом вы пройдете весь лабиринт от входа до выхода). Пройдя лабиринт, вы решаете пройти его еще раз (по тому же алгоритму с поворотом налево), но в этот раз начав с выхода и закончив на входе.

Ваш путь через лабиринт описывается тремя символами: 'W' означает, что вы двигаетесь вперед в следующую комнату, 'L' означает, что вы поворачиваете налево на 90 градусов, и 'R'

означает, что вы поворачиваете направо на 90 градусов. Вы начинаете свой путь снаружи лабиринта, на клетке прямо перед входом, лицом ко входу. Вы заканчиваете путь когда выходите из лабиринта через выход. Например, для представленного на рисунке лабиринта ваш путь будет следующим (вход на северной стороне, выход на западной):

WRWWLWWLWWLWLRWRWWWWRWWRLW



При обратном проходе от выхода ко входу ваш путь будет следующим:

WWRLWLWWLWWLWWRLWWRLW.

На основе двух строк, описывающих путь через лабиринт (в прямом и обратном направлении), ваша программа должна сгенерировать описание лабиринта

Входные данные

Программа должна считывать данные из текстового файла.

Первая строка файла содержит целое число N, описывающее количество заданий в файле. Далее следует N строк, описывающих прямое и обратное прохождение лабиринтов в формате:

прямое_прохождение обратное_прохождение

Все пути состоят как минимум из двух символов, содержат только символы 'W', 'L', 'R', начинаются и заканчиваются символом 'W'.

Выходные данные

Ваша программа должна выводить результат в текстовый файл.

Для каждого задания (пары строк с прямым и обратным прохождением) выведите строку 'Case #x', где x – номер задания. Следующие R строк файла должны описывать лабиринт размером R на C. В каждой из строк должно быть C символов, описывающих, в каких направлениях можно двигаться из этой комнаты. Используйте следующую таблицу для кодирования направлений:

Символ	Проход север?	наПроход юг?	наПроход запад?	наПроход восток?	на
1	Да	Нет	Нет	Нет	
2	Нет	Да	Нет	Нет	
3	Да	Да	Нет	Нет	
4	Нет	Нет	Да	Нет	
5	Да	Нет	Да	Нет	
6	Нет	Да	Да	Нет	
7	Да	Да	Да	Нет	
8	Нет	Нет	Нет	Да	

9	Да	Нет	Нет	Да
a	Нет	Да	Нет	Да
b	Да	Да	Нет	Да
c	Нет	Нет	Да	Да
d	Да	Нет	Да	Да
e	Нет	Да	Да	Да
f	Да	Да	Да	Да

Пример

Входной файл

```
2
WRWWLWWLWWLWLWRRWRWWWRRWLW  WRRWLWLWWLWWLWRRWRWWLW
WW WW
```

Выходной файл

```
Case #1:
ac5
386
9c7
e43
9c5
Case #2:
3
```

Тестовые данные

Для тестирования вашего приложения используйте следующие тестовые файлы:

- [Тестовый набор 1](#)
- [Тестовый набор 2](#)