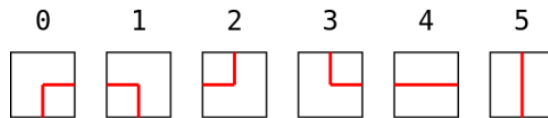


Задача А. Гирлянда

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 15 секунд
Ограничение по памяти: 256 мегабайт

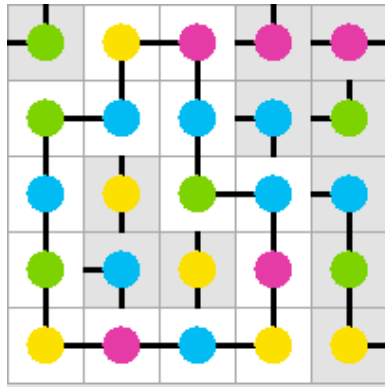
Вам дано мультимножество из $h \times w$ плиток размера 1×1 . Каждая плитка содержит диод определённого цвета в центре и два провода, каждый из которых соединяет диод с серединой какой-то стороны. Плитки нельзя поворачивать.

Цвета диодов кодируются латинскими буквами от a до d . Конфигурации проводов кодируются так: 0 — вправо и вниз, 1 — влево и вниз, 2 — влево и вверх, 3 — вправо и вверх, 4 — влево и вправо, 5 — вверх и вниз.



Ваша задача — расположить данные плитки в прямоугольнике высоты h и ширины w , так чтобы получилась гирлянда как можно большей длины. Гирлянда это цикл из плиток, такой что для каждого двух соседних плиток в этом цикле выполнены условия:

- они имеют общую сторону;
- в каждой из них есть провод, соединяющий диод с серединой этой общей стороны;
- их диоды имеют разный цвет.



Формат входных данных

В первой строке содержатся три числа h , w , c — высота и ширина прямоугольника и количество цветов. В следующей строке содержится $h \times w$ описаний плиток, каждое описание — это буква, означающая цвет, и цифра, означающая расположение проводов.

Формат выходных данных

Выведите h строк, в каждой из которых выведите w описаний плиток в том же формате, что и во входных данных. Мультимножество плиток должно совпадать со входными данными.

Система оценки

Результат на тесте равен длине гирлянды (максимального корректного цикла) в вашем ответе, разделённой на площадь прямоугольника и умноженной на 1000.

Пример

стандартный ввод
5 5 4 5b 2d 5a 2a 4a 4a 1a 0c 3c 1b 2d 5c 1b 1b 0d 2c 1b 3d 5b 4b 5c 5d 3c 5d 2b
стандартный вывод
2d 0c 1a 2a 4a 0d 2b 5b 1b 2d 5b 5c 3d 1b 1b 5d 1b 5c 5a 5d 3c 4a 4b 2c 3c

Замечание

Числа h и w будут выбраны случайно и равномерно в интервале от 10 до 100, а число c — случайно и равномерно в интервале от 2 до 4. Каждая плитка будет выбрана случайно и равномерно из всех возможных $6c$ типов плиток.

Визуализатор

Вам доступен визуализатор, который позволяет генерировать тесты и визуализировать их.

```
java -jar GarlandOfLightsVis.jar gen 566
```

генерирует в выходной поток тест с `seed=566`

```
java -jar GarlandOfLightsVis.jar vis 566.out
```

визуализирует ответ, выведенный в файл `566.out`, в виде картинки в файле `566.out.png`

Задача В. Уборка снега

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	15 секунд
Ограничение по памяти:	256 мегабайт

Вы открываете свой бизнес по уборке снега. У вас контракт с небольшим северным городом на 2000 дней.

Город можно упрощенно представить как квадратное поле со стороной b . Строки и столбцы будем нумеровать от 0 до $b - 1$ сверху вниз и слева направо соответственно.

Каждая клетка может быть чистой или содержать снег. Перед первым днём вашей работы все клетки чистые. Затем каждый день (рано утром) в каких-то клетках может выпасть снег.

Чтобы чистить снег, вы нанимаете дворников. Их может быть не более 100. Увольнять их нельзя. Когда вы нанимаете нового дворника, вы можете выбрать, где его расположить. Дворник появится в выбранной клетке, и если в ней был снег, то к концу дня клетка станет чистой. Начиная со следующего дня каждый день вы можете оставлять дворника на месте или передвигать в любую соседнюю по стороне клетку. После перемещения или оставления на месте дворник сделает чистой клетку, на которой он оказался, если там был снег.

Обратите внимание, что дворник сначала перемещается, а потом чистит снег, то есть клетка, с которой дворник уходит, остаётся в этот день непочищенной.

В одной клетке может одновременно находиться несколько дворников.

В конце каждого дня каждому дворнику надо заплатить зарплату s , а за каждую неубранную от снега клетку — штраф городу f .

За 2000 дней работы вы получите фиксированную сумму, то есть чтобы максимизировать прибыль, нужно минимизировать расходы на зарплаты и штрафы. Это и есть ваша задача.

Протокол взаимодействия

Сначала вы считываете три целых числа b , s , f — размер поля, зарплату дворника и штраф за неубранную клетку.

Затем вы 2000 раз повторяете процедуру: считываете информацию о выпавшем рано утром снеге и выводите информацию о ваших действиях в этот день.

Информация о выпавшем снеге представляет собой: в первой строке количество клеток, в которых выпал снег k , а во второй строке $2k$ чисел через пробел — координаты клеток, на которые выпал снег. Каждая клетка описывается как номер строки, затем номер столбца. Если $k = 0$, то вторая строка пустая.

Информация о ваших действиях представляет собой: в первой строке количество действий в этот день, и затем соответствующее количество строк вида «N r c» и «M id dir».

Строка «N r c» означает, что вы нанимаете дворника и располагаете его в строке r и столбце c . Он получает первый свободный идентификатор, начиная с нуля. Если в один день вы наняли несколько дворников, они получают последовательные идентификаторы в порядке обработки ваших действий сверху вниз.

Строка «M id dir» означает, что вы передвигаете дворника с идентификатором id (нанятого не в этот день, а ранее) в направлении dir . Направление может быть 'U' (вверх), 'D' (вниз), 'L' (влево) или 'R' (вправо).

Ещё раз обозначим порядок обработки одного дня: сначала выпадает снег, затем выполняются найм и перемещения дворников, затем все дворники чистят клетки, на которых стоят, и затем вычисляются ваши расходы на зарплаты и штрафы городу.

Система оценки

Пусть $dumb$ — расходы, которая получила бы в том же городе фирма, которая вообще не нанимает дворников, а $your$ — ваши расходы. Результат на тесте равен $\max(0, 1000 \cdot (1 - your/dumb)^3)$.

Замечание

b выбирается случайно и равномерно в интервале между 20 и 50, s и f выбираются независимо, случайно и равномерно в интервале между 10 и 100.

Снег выпадает из движущихся туч. Вы можете запустить визуализатор или изучить его исходный код, чтобы разобраться в используемой модели.

Визуализатор

Вам доступен визуализатор, который позволяет показывать процесс уборки снега.

`java -jar SnowCleaning.jar -seed 566 -exec "mysolution.exe tra la la"` показывает работу вашего решения, запущенного командой «`mysolution.exe tra la la`», на тесте с `seed=566`

`java -jar SnowCleaning.jar -seed 566 -novis -exec "mysolution.exe tra la la"` то же самое, но без визуализации, только вывод счёта

`java -jar SnowCleaning.jar -seed 566` показывает выпадение снега на тесте с `seed=566` (как будто вы всегда выводите 0).