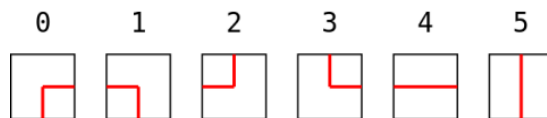


Гирлянда

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

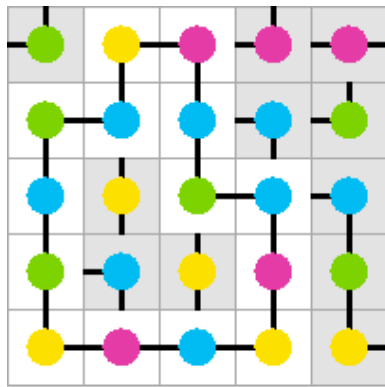
Вам дано мультимножество из $h \times w$ плиток размера 1×1 . Каждая плитка содержит диод определённого цвета в центре и два провода, каждый из которых соединяет диод с серединой какой-то стороны. Плитки нельзя поворачивать.

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 пропорционален длине гирлянды (максимального корректного цикла) в вашем ответе.

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 равновероятно в интервале от 10 до 100, а число c — случайно и равновероятно в интервале от 2 до 4. Каждая плитка будет выбрана случайно и равновероятно 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`