

## Задача А. 38 попугаев

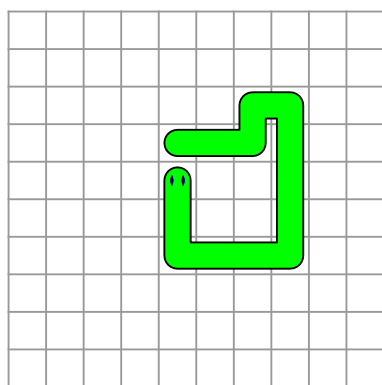
Имя входного файла:	<code>stdin</code>
Имя выходного файла:	<code>stdout</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Как-то раз удав, слонёнок, мартышка и попугай играли в замечательную игру «змейка».

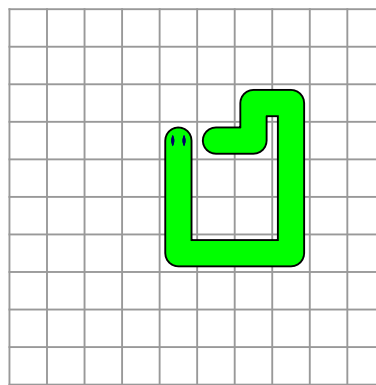
Суть этой известной игры в следующем. Удав перемещается по прямоугольному полю, не переползая через самого себя, и ест яблоки, которые ему сверху скидывает попугай.

Как известно, мерять длину удава можно в слонёнках, мартышках или попугаях. Но в попугаях удав выходит гора-а-а-здо длиннее, поэтому мы остановимся на них в качестве единицы измерения. Поле представляет собой  $N$  строк по  $M$  ячеек размером 1 попугай  $\times$  1 попугай. Изначально длина удава составляет всего три попугая (каким образом удав становится таким коротким в начале игры — долгая и запутанная история, поэтому мы не будем на ней останавливаться в рамках этой задачи. . . ), он занимает три крайних левых ячейки в верхней строке поля и смотрит направо. Каждый раз, съедая яблоко, удав удлиняется на одного попугая и занимает тем самым ещё одну клетку поля.

Более формально — удав длины  $L$  попугаев представляет собой последовательность из  $L$  различных клеток поля  $A_1, A_2 \dots A_L$ , таких что любые две соседних в этой последовательности клетки  $A_i$  и  $A_{i+1}$  являются соседними на поле, т. е. имеют общую сторону. Головой удава считается клетка  $A_L$ . За один ход удав может продвинуться на одну клетку — он выбирает новую клетку  $H$  для своей головы (естественно, смежную с текущей клеткой головы  $A_L$  и не выходящую за границы поля), а его хвост подтягивается следом на одну клетку. То есть, после хода удав будет представлять собой последовательность из клеток  $A_2, A_3, \dots, A_L, H$ . **Обратите внимание**, по определению выше удав после выполнения хода не должен дважды содержать одну и ту же клетку, но новое положение головы  $H$  вполне может совпадать со старым положением хвоста  $A_1$  (наш удав перемещается подобно гусенице, сначала сжимаясь и подтягивая хвост, а потом расжимаясь и продвигая голову, поэтому это корректный ход).



вот так...



...ходить можно

Заметим, что из этого определения сразу следует, что в каждый момент времени у удава есть возможность пойти в не более чем трёх направлениях — продолжить движение в направлении, куда смотрит его голова, свернуть налево или свернуть направо.

В любой момент времени на поле находится ровно одно яблоко. Каждый раз, когда удав съедает яблоко, его длина увеличивается на одного попугая. Это происходит следующим образом — если удав делает ход, после которого его голова занимает клетку, в которой расположено яблоко, оно считается съеденным. В таком случае яблоко сразу продвигается по телу удава вплоть до бывшего положения хвоста (т. е. до клетки  $A_1$ ), и на этом месте возникает новая клетка удава. Таким образом, удав приобретает длину в  $(L+1)$  попугай и занимает к началу следующего хода клетки  $(A_1, A_2, \dots A_K, H)$ . В этот же момент попугай скидывает следующее яблоко на какую-то из незанятых клеток поля.

Удав проигрывает, если он делает некорректный ход. Помогите удаву съесть все яблоки!

---

## Формат взаимодействия с тестирующей системой

Это — интерактивная задача.

В первой строке стандартного потока ввода идут два целых числа  $N$  и  $M$  ( $3 \leq N, M \leq 25$ ) — количество строк и столбцов поля соответственно.

Далее ваше общение с тестирующей программой будет устроено следующим образом. На каждое яблоко, скидываемое попугаем, вашей программе будут подаваться на стандартный на стандартный поток ввода два целых числа  $R$  и  $C$  ( $1 \leq R \leq N, 1 \leq C \leq M$ ) — соответственно номер строки и столбца клетки, содержащей яблоко. Строки нумеруются с 1 по  $N$  сверху вниз, столбцы нумеруются с 1 по  $M$  слева направо.

После этого вы должны вывести последовательность команд для удава в виде строки, состоящей из латинских букв 'R', 'U', 'L' и 'D', которые обозначают что удав должен сделать ход, передвинув свою голову соответственно на клетку правее, выше, левее или ниже. По итогам этой последовательности команд удав должен съесть яблоко, не допуская некорректных ходов. **Обратите внимание**, яблоко должно быть съедено именно последним ходом этой последовательности! После вывода каждой строки сбрасывайте буфер вывода — для этого используйте `flush(output)` на языке Паскаль или Delphi, `fflush(stdout)` или `cout.flush()` в C/C++, `sys.stdout.flush()` на языке Python, `System.out.flush()` на языке Java.

В момент окончания игры (как успешного, так и неуспешного — например, если удав делает некорректный ход) вашей программе на ввод подаётся строка, состоящая из двух нулей. **Обратите внимание**, считав эти два нуля, ваша программа обязательно должна **сразу** завершиться! В противном случае в качестве результата проверки вы можете получить вердикт, отличный от настоящего!

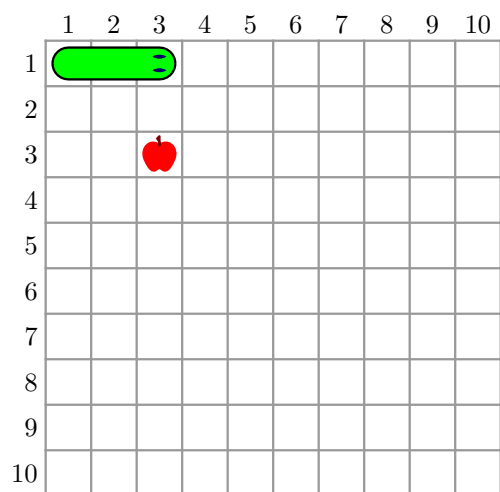
### Примеры

stdin	stdout
4 6 3 3	DD
4 6	DRRR
1 3	UUULLL
0 0	
3 5 2 3	D
3 4	DR
3 5	R
2 2	ULLL

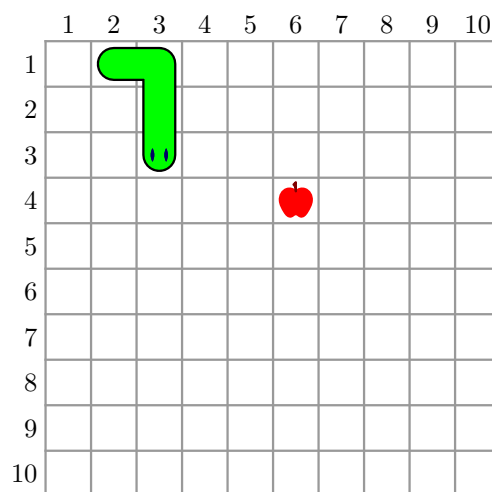
### Note

Пояснение к первому тесту. Положение удава по ходу игры показано на следующей странице.

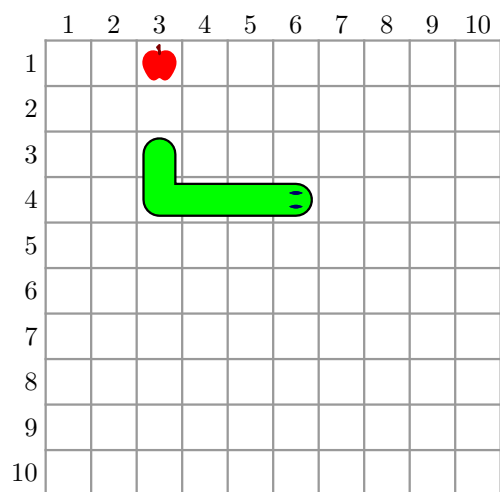
Пояснение ко второму тесту. Обратите внимание, что после шестого хода происходит ситуация, проиллюстрированная в условии: в качестве нового положения головы выбирается клетка (2,3), являющаяся на тот момент клеткой хвоста.



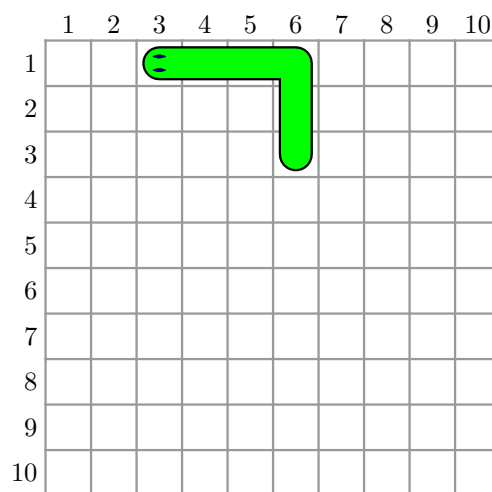
до начала игры



после 2 хода



после 6 хода



после 12 хода