

Задача А. Короля — в угол

Имя входного файла: `king1.in`
Имя выходного файла: `king1.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

В левом нижнем углу доски $m \times n$ стоит король. Двое игроков по очереди ходят королем, перемещая его на одну клетку по вертикали вверх, по горизонтали вправо или по диагонали вправо-вверх. Проигрывает тот, кто не может сделать ход. Определите, какой из игроков имеет выигрышную стратегию.

Формат входных данных

На вход программе подается два натуральных числа m и n , не превосходящих 10^9 .

Формат выходных данных

Программа должна вывести номер игрока (1 или 2), который имеет выигрышную стратегию.

Примеры

<code>king1.in</code>	<code>king1.out</code>
3 4	1

Задача В. Кони ходят по очереди

Имя входного файла: `knight3.in`
Имя выходного файла: `knight3.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Переведите каждого из двух коней из одной клетки в другую за наименьшее общее число ходов. Два коня не могут одновременно находиться в одной клетке. **Ходы коней должны чередоваться.**

Формат входных данных

Во входном файле записаны координаты первого и второго коня, затем координаты клеток, куда нужно их переместить.

Формат выходных данных

Программа должна вывести последовательность ходов коней в виде нескольких строк. Первым символом в строке должен быть номер коня (1 или 2), затем, через пробел, координаты клетки, в которую он переставляется. Необходимо вывести любое из возможных оптимальных решений. Кони должны ходить по очереди, первым может ходить любой из коней, кони могут сделать различное число ходов.

Примеры

<code>knight3.in</code>	<code>knight3.out</code>
<code>a1</code>	<code>1 b3</code>
<code>c2</code>	<code>2 a1</code>
<code>c2</code>	<code>1 d4</code>
<code>a1</code>	<code>2 b3</code>
	<code>1 c2</code>
	<code>2 a1</code>

Задача С. Лодейное поле

Имя входного файла: `rooks.in`
Имя выходного файла: `rooks.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

На шахматной доске 8×8 стоит белый король и несколько черных ладей. Найдите безопасный ход, который может сделать король, либо определите, что королю поставлен мат или пат.

Формат входных данных

В первой строке содержится позиция белого короля. Во второй строке записано число n ($0 \leq N \leq 63$) — количество черных ладей на поле. В последующих N строках записаны позиции ладей.

Формат выходных данных

Выведите слово `Checkmate`, если белому королю поставлен мат, слово `Stalemate`, если позиция патовая, или произвольный безопасный ход белого короля в противном случае.

Примеры

<code>rooks.in</code>	<code>rooks.out</code>
a1 2 b8 h2	Stalemate
e4 4 e3 e5 d4 f4	Checkmate
h8 1 g7	g7

Задача D. Мирные ферзи — 2

Имя входного файла: `queen3.in`
Имя выходного файла: `queen3.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

По данному n определите количество способов расставить n ферзей на доске $n \times n$, если считать совпадающими расстановки, совпадающие при поворотах и отражениях доски.

Формат входных данных

Программа получает на вход натуральное число n , не превосходящее 10.

Формат выходных данных

Программа должна вывести единственное число.

Примеры

<code>queen3.in</code>	<code>queen3.out</code>
2	0
4	1

Задача Е. Доминирующие ферзи

Имя входного файла: `queen4.in`
Имя выходного файла: `queen4.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Известно, что на шахматной доске 8×8 можно расставить 5 доминирующих ферзей, то есть расставить их так, чтобы каждая клетка шахматной доски была либо занята ферзем, либо находилась под боем одного из ферзей.

Дана клетка шахматной доски. Определите количество способов расставить 4 ферзей по оставшимся клеткам, чтобы вместе с ферзем, стоящим в исходной клетке, они образовывали бы доминирующую систему.

Формат входных данных

Программа получает на вход координаты одной клетки.

Формат выходных данных

Программа должна вывести единственное число: число искомых расстановок.

Примеры

<code>queen4.in</code>	<code>queen4.out</code>
a1	482

Замечание

Вместо `python3` для сдачи этой задачи используйте `руру3`. Обратите внимание, что ограничение по времени для языка `python3` и `руру3` составляют 6 секунд.

Задача F. Обход доски

Имя входного файла:	–
Имя выходного файла:	–
Ограничение по времени:	–
Ограничение по памяти:	–

Дана шахматная доска размером $n \times m$. Необходимо построить обход всей доски ходом коня так, чтобы конь побывал во всех клетках доски ровно по одному разу. В задачах F1–F6 достаточно построить незамкнутый путь, а в задачах F7–F9 необходимо построить циклический обход, начальная и конечная клетка которого также должны быть связаны ходом коня.

Вы должны сдать на проверку текстовый файл, содержащий nm строчек. Каждая строчка должна содержать координаты ровно одной клетки. Две соседние координаты должны быть связаны ходом коня и каждая из nm клеток доски должна встречаться в этом файле ровно один раз. Каждая клетка записывается в виде “a1”, где сначала записывается одна из первых n букв латинского алфавита, затем — число от 1 до m . Например, для доски 4×5 сданный файл может быть таким:

```
a1
c2
d4
b5
a3
b1
d2
c4
a5
b3
c1
a2
b4
d5
c3
d1
b2
a4
c5
d2
```

Список файлов для сдачи:

- F1: путь на доске 5×5 ;
- F2: путь на доске 7×7 ;
- F3: путь на доске 6×12 ;
- F4: путь на доске 13×14 ;
- F5: путь на доске 20×20 ;
- F6: путь на доске 26×26 ;
- F7: цикл на доске 6×6 ;
- F8: цикл на доске 8×8 ;
- F9: цикл на доске 7×16 .

Кроме этого, сдайте программу, которую вы использовали для нахождения ответов. Эта программа не будет проверяться автоматически. Сдать эту программу необходимо как задачу “F”.