

BEARTRAP: Bear and Cat Trap

Легенда, описанная далее, переформулирована и упрощена переводчиком, чтобы читатель мог лучше понять условие задачи. Оригинальную легенду вы можете прочитать на странице задачи в контексте.

Условие:

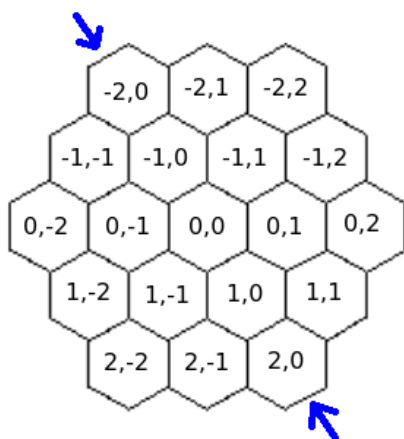
Это интерактивная задача. Это значит, что каждый раз, когда вы выводите данные, необходимо вывести символ новой строки и очистить поток выходных данных. Например, на C++ это делается с помощью команды `fflush(stdout)`, в Java - функция `System.out.flush()`, в Python - функция `sys.stdout.flush()`, а в Pascal — `flush(output)`. Только после очистки потока можно прочесть входные данные.

Лимак - маленький полярный мишка. Одна из его любимых игр - это Cat Trap (еще известна как Black Cat, Chat Noir или Circle The Cat). Цель игры - не дать коту убежать за пределы поля. Вы можете поиграть в эту игру сами: <http://www.gamedesign.jp/flash/chatnoir/chatnoir.html>. Еще Лимак очень любит кошек, но они его очень боятся (без видимой на то причины) и не хотят играть с Лимаком.

В этой задаче Вы должны пройти модифицированную версию данной игры. Отличия от оригинала Cat Trap следующие:

- Игровое поле - правильный шестиугольник со стороной $N = 20$.
- Изначально ни одна из клеток не заблокирована.
- Необходимо выиграть игру за M ходов (заблокировав не более M клеток).
- Кот будет следовать стратегии, описанной далее.

Ряды игрового поля пронумерованы от $-(N-1)$ до $(N-1)$ сверху вниз, нулевой ряд проходит ровно посередине. Столбцы пронумерованы от $-(N-1)$ до $(N-1)$ слева направо, нулевой столбец - это диагональ, которая содержит первую клетку первого ряда и последнюю клетку последнего ряда. Например, для $N = 3$ игровое поле имеет следующий вид (синие стрелки указывают на нулевой столбец):



Пусть (r, c) обозначает клетку поля на r -том ряду в c -том столбце. Эта клетка имеет шесть смежных клеток: $(r, c+1)$, $(r-1, c+1)$, $(r-1, c)$, $(r, c-1)$, $(r+1, c-1)$, $(r+1, c)$.

Граничная клетка - это клетка с меньшим чем 6 количеством соседних клеток (всего $6 \cdot (N-1)$ граничных клеток на поле). Для любой незаблокированной клетки X определим **расстояние до границы** как минимальное количество ходов, которое потребуется коту, чтобы достичь граничной клетки, если изначально он находится в

клетке X и никакие дополнительные клетки не будут заблокированы. Расстояние до границы считается равным бесконечности, если невозможно добраться до граничных клеток.

Изначально кот находится в клетке $(0, 0)$, т.е. в середине игрового поля. Каждый ход игры происходит в следующем порядке:

- Кот ходит первым. Если он находится на граничной клетке, он убегает и Вы проигрываете. Если он не может достичь граничной клетки, то вы выигрываете (т.е. кот находится в связной компоненте клеток, окруженных заблокированными клетками). Иначе кот переходит в соседнюю клетку, у которой наименьшее расстояние до границы. Например, пусть три смежные клетки заблокированы, расстояние до границы для двух из этих клеток равно 15, а у оставшейся расстояние до границы равно 17. Тогда кот выберет одну из клеток с расстоянием 15. На первом ходе кот выбирает случайным образом одну из смежных клеток.
- Затем Вы можете заблокировать одну из клеток. Заблокировать можно только пустые клетки, т.е. те, которые уже не заблокированы и в них нет кота.

Взаимодействие с тестирующей программой:

Первая строка входных данных содержит разделенные пробелом целые числа T и M - количество тестовых случаев и максимальное возможное количество раз, которое можно заблокировать клетку (для каждого тестового случая). Для каждого тестового случая изначально кот находится в клетке $(0, 0)$ и все клетки разблокированы.

После хода кота вы получаете информацию в виде строки: "WIN" либо "CAT r c" (без кавычек). Строка "WIN" означает, что кот не может достичь граничных клеток и Вы победили. Строка "CAT r c" означает, что кот делает ход в клетку (r, c) .

Затем необходимо вывести информацию о Вашем ходе в формате "BLOCK r c" (без кавычек), обозначающих блокирование клетки (r, c) . Не забывайте выводить символ переноса строки и очищать поток выходных данных. Отметим, что ход пропускать нельзя, и нельзя заблокировать уже заблокированную клетку или клетку с котом.

Подзадачи:

- Подзадача 1 (35 баллов) $1 \leq T \leq 10, M = 500$
- Подзадача 2 (65 баллов) $1 \leq T \leq 100, M = 20$

Визуализация:

Вам необходимо установить Python3 для запуска визуализатора. Подобные инструкции и пример использования Вы можете найти по этой [ссылке](#).

Примеры тестов:

Ваш решение	Тестирующая программа
	2500
	CAT 1 -1
BLOCK -1 0	
	CAT 1 0
BLOCK -1 1	
	CAT 0 0

Ваш решение	Трестирующая программа
BLOCK 1 -1	
	CAT 0 1
BLOCK 0 -1	
	CAT 1 0
BLOCK 1 1	
	CAT 0 0
BLOCK 0 2	
	CAT 0 1
BLOCK -1 2	
	CAT 0 0
BLOCK 1 0	
	WIN
	CAT 1 0
BLOCK 5 -12	
...	...
	WIN

Пояснения:

В данном примере $T = 2$ тестовых случая, в каждом из которых разрешено заблокировать не более $M = 500$ клеток. Отметим, что пример некорректный, т.к. кот не всегда ходил в смежные клетки с минимальным расстоянием до границы. Второй тест добавлен для того, чтобы продемонстрировать взаимодействие с трестирующей программой при нескольких тестовых случаях.

Вы можете запустить визуализатор для первого теста. Для Вашего удобства игровое поле показывает финальное состояние поля. Предположим, что у данного поля $N = 4$ (для все тестовых случаев $N = 20$).

```

. / . /
/ . / . /
. # # # ,
, . # с , # ,
, . # # # .
/ . / . /
, . / .

```