

Bear and Cat Trap

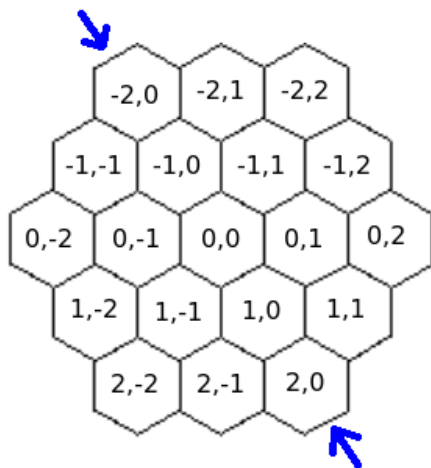
Đây là một bài toán tương tác. Có nghĩa là bất cứ khi nào bạn in ra một thứ gì đó, bạn phải kết thúc với ký tự dòng mới và flush output. Ví dụ, ở C++ sử dụng hàm `fflush(stdout)`, trong Java là `System.out.flush()`, trong Python là `sys.stdout.flush()`, và trong Pascal là `flush(output)`. Chỉ sau khi flush the output thì bạn mới có thể đọc lại input.

Limak là một chú gấu Bắc Cực nhỏ xinh. Trò chơi yêu thích của anh ta là Bẫy mèo (hay còn gọi là Mèo đen, Chat Noir hoặc Circle The Cat). Mục đích của trò chơi là ngăn con mèo không cho nó chạy đi. Limak yêu thích trò chơi này bởi vì mèo thường sợ anh (không lí do nào cả) và không muốn chơi với anh ấy. Bạn có thể thử trò chơi này tại đây: <http://www.gamedesign.jp/flash/chatnoir/chatnoir.html>.

Trong bài toán này, bạn cần chiến thắng trong một phiên bản trò chơi sửa đổi. Điểm khác biệt của trò chơi mới này như sau:

- Lưới là một hình lục giác thường với kích thước $N = 20$.
- Ban đầu không có ô nào bị chặn.
- Bạn phải chiến thắng trong M lượt (cụ thể: bạn phải chặn được ít nhất là M ô).
- Mèo thường có một chiến thuật di chuyển đặc biệt như mô tả dưới đây:

Các hàng được đánh số từ $-(N-1)$ đến $N-1$ từ đỉnh đến đáy. Do vậy hàng 0 ở giữa. Các cột được đánh số từ $-(N-1)$ đến $N-1$ từ trái sang phải, khi đó cột 0 là đường chéo nối ô đầu tiên của hàng cao nhất và ô cuối cùng của hàng thấp nhất. Ví dụ: $N=3$ thì lưới có thể miêu tả như sau: (mũi tên màu xanh đang chỉ vào cột 0)



Cho (r, c) là ô tại vị trí giao nhau của hàng thứ r và cột thứ c . Các mô tả trên cho thấy rằng ô (r, c) có 6 ô gần nhau. $(r, c+1)$, $(r-1, c+1)$, $(r-1, c)$, $(r, c-1)$, $(r+1, c-1)$, $(r+1, c)$.

Một **ô ở biên** là ô với tiếp giáp với ít hơn 6 ô khác. (có chính xác là $6*(N-1)$ ô biên). Với bất cứ ô X nào không bị chặn lại thì chúng ta gọi nó là **khoảng cách đến biên** – bởi vì số lượt di chuyển của con mèo để đến được cột biên là nhỏ nhất. Giả thiết rằng Limak đi từ X và không có ô nào bị chặn nữa. Theo đó thì khoảng cách đến biên là 0 vì tất cả các ô biên đều không bị chặn. Khoảng cách đến biên là vô hạn nếu như một ô bị chặn và con mèo không thể tiếp cận được bất cứ ô biên nào.

Con mèo bắt đầu ở ô $(0, 0)$, cụ thể là ở giữa của lưới. Trong mỗi lượt của trò chơi:

- Mèo di chuyển trước. Nếu nó trong một ô biên, anh ấy chạy mất thì bạn sẽ thua. Nếu như anh ấy không thể tiếp cận được một ô biên nào thì bạn thắng (cụ thể là khi con mèo nằm ở vị trí mà bị các ô khác bị chặn lại bao xung quanh) Ngược lại, mèo di chuyển đến một trong các ô liền kề với khoảng cách nhỏ nhất để đến biên. Ví dụ: nếu như có ba ô liền kề bị chặn, hai ô có khoảng cách đến biên là 15, một ô có khoảng cách đến biên là 17, sau đó con mèo lựa chọn hai ô với khoảng cách 15. Đương nhiên, trong lần đi đầu tiên con mèo chọn bất cứ ô nào trong 6 ô liền kề, bởi vì tất cả chúng đều có khoảng cách đến biên bằng nhau.
- Sau đó bạn chặn một ô lại. Bạn không thể chặn một ô nếu như nó vừa được chặn hoặc nó chứa con mèo.

Sự tương tác

Dòng đầu tiên của input chứa hai số nguyên **T** và **M** – lần lượt là số test và số lượt nhiều nhất mà bạn có thể chặn các ô lại. (cho mỗi test). Ban đầu mỗi test, con mèo ở vị trí ô $(0, 0)$ và không có ô nào bị chặn.

Khi con mèo được giả định là đang di chuyển, bạn nên đọc input, một dòng hoặc là “WIN” hoặc là “CAT $r\ c$ ” (không có ngoặc kép). Một dòng “WIN” có nghĩa là con mèo không thể tiếp cận được biên thêm nữa và bạn nên thử các test tiếp theo – sai đó bạn nên kết thúc chương trình của bạn. Dòng “CAT $r\ c$ ” có nghĩa là con mèo di chuyển đến ô (r, c) .

Sau khi con mèo di chuyển, bạn nên in ra một dòng với format “BLOCK $r\ c$ ” – có nghĩa là chặn cột (r, c) . Nhớ rằng in một dòng chữ cái mới và flushing the output. Bạn không thể bỏ 1 lượt đi nào và bạn không thể chặn một ô nếu ô đó đã bị chặn hoặc nó đang chứa con mèo.

Subtasks

- Subtask #1 (35 điểm) $1 \leq T \leq 10$, $M = 500$
- Subtask #2 (65 điểm) $1 \leq T \leq 100$, $M = 20$

Visualizer

Bạn cần Python3 để chạy **visualizer**: [link](#). Phần giới thiệu và cách sử dụng được nêu ở cuối file.

Ví dụ

your program	the system
--------------	------------

	2 500
	CAT 1 -1
BLOCK -1 0	
	CAT 1 0
BLOCK -1 1	
	CAT 0 0
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

Giải thích

Trong ví dụ đã cho, có $T = 2$ test và mỗi test bạn được cho phép chặn $M = 500$ ô. Lưu ý rằng ví dụ có những thông tin không hợp lý bởi vì con mèo không thể lúc nào cũng di chuyển đến một ô liền kề với khoảng cách ngắn nhất đến biên. Một tình huống không thể xảy ra trong test thực tế. Và test thứ hai được cho chỉ để bạn thấy được cách giao tiếp như thế nào nếu có nhiều test case.

Bạn có thể chạy visualizer để thấy sự mô phỏng test 1. Để tiện cho bạn, lưới dưới đây chỉ ra tình huống cuối cùng, giả thiết rằng kích cỡ của lưới là $N = 4$ (nhưng nhớ rằng $N = 20$ trong mọi test).

```

      . / . /
    / . / . /
  . / # # # /
 / . # c / # /
 / . # # # .
    / . / . /
      / . / .

```