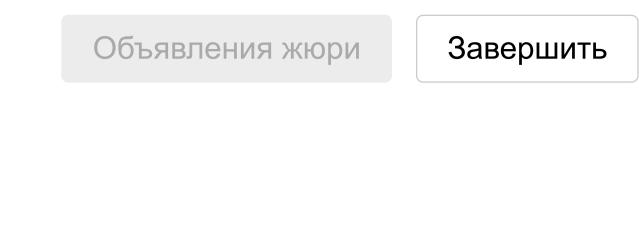


18 мар 2024, 20:08:08
 старт: 18 мар 2024, 19:22:28
 финиш: 18 мар 2024, 22:22:28
 до финиша: 02:14:14



А. Бесконечные

крестики-нолики

С. Каналы связи

В. Покупатель рыбы

Задачи Посылки Сообщения

А. Бесконечные крестики-нолики

Ограничение времени 2 секунды
Ограничение памяти 256Mb
Ввод стандартный ввод или input.txt
Вывод стандартный вывод или output.txt

Игра в крестики-нолики на бесконечной плоскости похожа на обычные крестики-нолики: два игрока по очереди ставят свои фигуры (крестики у первого игрока и нолики — у второго) в свободные клетки поля. Побеждает тот игрок, который первым выстроил пять своих фигур по горизонтали, вертикали или одной из диагоналей.

В логе записаны координаты клеток, в том порядке, в котором игроки ставили свои фигуры. Определите, кто победил в игре или отследите ситуацию, что игроки увлеклись и продолжили игру после победы одного из игроков.

Формат ввода

В первой строке записано число n ($1 \le n \le 10000$) — количество ходов, которые совершили игроки.

В следующих n строках записано по два числа r, c (|r|, $|c| \le 10^9$) — координаты клетки, в которую была поставлена очередная фигура. Гарантируется, что все координаты клеток уникальны (т.е. игрок не ставил свою фигуру в ту клетку, в которой уже стоит фигура)

Формат вывода

В случае победы первого игрока последним ходом выведите слово "First". В случае победы второго игрока последним ходом выведите слово "Second". Если ни один из игроков не успел победить выведите слово "Draw". Если ходы продолжились после победы одного из игроков выведите слово "Inattention".

Пример 1

Ввод	Вывод	
9	First	
4 4		
4 5		
2 2		
2 3		
3 3		
3 4		
1 1		
1 2		
5 5		

Пример 2

Ввод	Вывод 🗇
10	Inattention
5 0	
1 1	
4 0	
2 1	
3 0	
3 1	
2 0	
4 1	
1 0	
5 1	

Язык Python 3.9 (PyPy 7.3.11) ~

Набрать здесь Отправить файл

```
1 n = int(input())
    coords = set()
    steps = [tuple(map(int, input().split())) for _ in range(n)]
    res = "Draw"
   9 def gen_cell(pos, center_row, center_col, dir_r, dir_c):
           [] = []
           for i in range(1, pos + 1):
    lst.append((center_row - dir_r * i, center_col - dir_c * i))
          for i in range(5 - pos):
    lst.append((center_row + dir_r * i, center_col + dir_c * i))
           return lst
def check_win(center_row, center_col, figure):
    for dir_r, dir_c in ((1, 0), (0, 1), (1, 1), (-1, 1)):
        for pos in range(5):
        cells = gen_cell(pos, center_row, center_col, dir_r, dir_c)
        if all((x, y, figure) in coords for x, y in cells):
            return True
           return False
for i, (r, c) in enumerate(steps):
player = i % 2
          coords.add((r, c, player))
if check_win(r, c, player):
   if i == n - 1:
                        res = "First" if i % 2 == 0 else "Second"
                 else:
                        res = "Inattention"
```

🗓 осталось 98 попыток

Время посылки	ID Задача				Следу	⁄ющая				
		Компилятор	Вердикт	Тип посылки	Время	Память	Тест	Баллы		
18 мар 2024, 19:51:24	110060292	Α	Python 3.9 (PyPy 7.3.11)	OK	-	446ms	29.87Mb	-	_	отчё-
18 мар 2024, 19:46:00	110059793	Α	Python 3.9 (PyPy 7.3.11)	WA	-	209ms	28.09Mb	1	-	отчё

Справка Обратная связь Пользовательское соглашение © 2013–2024 ООО «Яндекс»