

This is another adhoc game where you have identify the winning and losing positions. In the base case, you can mark the cells from which you can't move to any other cells as losing positions.

Next, for each cell, try to see if you can move to a losing position. If YES, then you can force your opponent into a losing state and win the game. If NO, then you are in a losing state and won't be able to win the game.

Don't perform the same calculation twice, simply perform them once and store the values in a table. The time complexity is $O(n^2)$, where n is the dimension of the table.



Set by [Shafaet](#)

Problem Setter's code :

Python 2

```
import sys
#sys.stdin=open("in","r")
#sys.stdout=open("out","w")
dic={}
def call(x,y):

    if x<1 or y<1 or x>15 or y>15:
        return 1
    if (x,y) in dic:
        return dic[(x,y)]
    state=1
    state = state and call(x-1,y-2)
    state = state and call(x+1,y-2)
    state = state and call(x-2,y+1)
    state = state and call(x-2,y-1)
    dic[(x,y)]=int(state==0)
    return dic[(x,y)]

t=int(raw_input())

for _ in range(t):
    x,y=map(int,raw_input().split())
    if call(x,y)==0:
        print "Second"
    else:
        print "First"
```