Lighting Trouble

DiPS CodeJam 22

Prompt

You're organising a code conference for your school this year, and the volunteers have messed up the lighting. Special lamps are used to light the conference. All the stalls are arranged in a square grid of size $n \times n$, and a lamp covers an entire square. A lamp will light up all stalls in the same row and column. Your volunteers were not organized enough to realize where to place the remaining lamps. Your job, given the position of the lamps placed so far, is to find out how many stalls have not been lit.

Input Format

- The first line of input contains a single integer n, denoting the size of the $n \times n$ grid.
- The next line contains an integer m, denoting the number of lamps used.
- The next m lines of input contain 2 integers, giving the row and column of each lamp in the format (x, y).

Output Format

Your output should contain a single integer, denoting the number of stalls (or grid squares) that are not lit.

Constraints

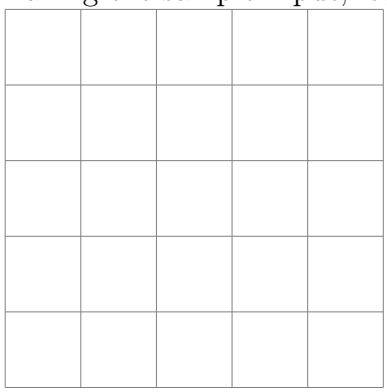
- $2 \geq n \geq 100$
- $1 \geq m \geq 5000$

Sample Input/Output

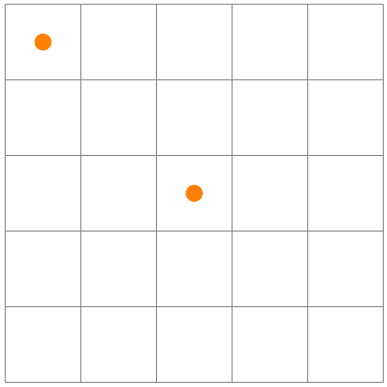
Input	Output
5 2 1 1 3 3	9

Solution

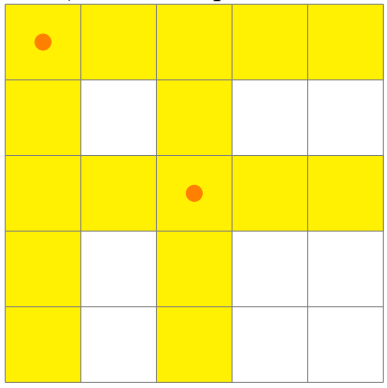
Taking the sample input, let's create a 5×5 grid.



Now, let's place the lamps. Taking the sample input, let's place the lamps at $(1, 1)$ and $(3, 3)$.



Now, let's fill up the stalls that are lit by the lamps:



We see that there are 9 stalls (grid squares) that are not lit.

Sample Program

```
n = int(input()) # Line 1 input
m = int(input()) # Line 2 input

# Helper functions:
def fillXOfArray(arr, n):
    newArr = arr;
    for i in range(len(arr[n])):
        arr[n][i] = 1
    return newArr
```

```
def fillYOfArray(arr, n):
    newArr = arr;
    for i in range(len(arr)):
        arr[i][n] = 1
    return newArr

arrx = []
arry = []
for i in range(0, m):
     m_input = list(map(int, input().split(" ")))
     arrx.append(int(m_input[0])) # x inputs from stdin
     arry.append(int(m_input[1])) # y inputs from stdin

grid = [[None] * n for i in range(n)] # Grid of size n

for i in range(len(arrx)):  # for each value of x,y (arrx and arry are the same length)
     grid = fillXOfArray(grid, arrx[i]-1) # fill that row with 1 (denoting lighted
   stalls)
     grid = fillYOfArray(grid, arry[i]-1) # fill that column with 1 (denoting lighted
   stalls)

sumOfOccurences = 0;
for i in range(len(grid)):
     sumOfOccurences+=sum(x is None for x in grid[i]) # Find the number of stalls that
   are not lit

print(sumOfOccurences)
```