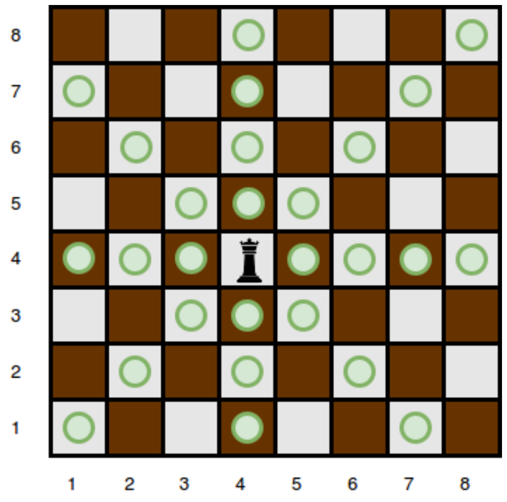
Queen's Attack II

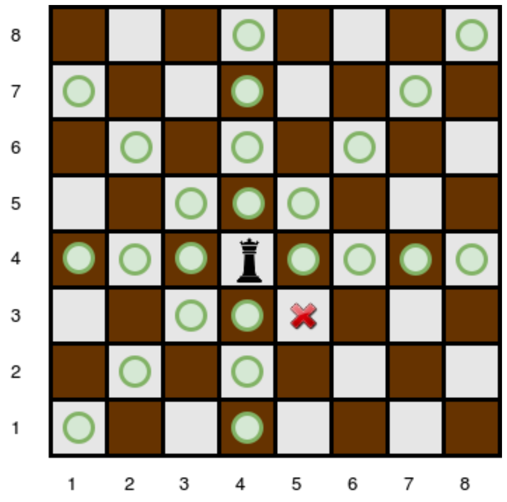
Author	bishop15
Difficulty	Medium
Max Score	30
Submitted By	11780

A queen is standing on an $n \times n$ chessboard. The chessboard's rows are numbered from 1 to n, going from bottom to top; its columns are numbered from 1 to n, going from left to right. Each square on the board is denoted by a tuple, (r, c), describing the row, r, and column, c, where the square is located.

The queen is standing at position (r_q, c_q) and, in a single move, she can attack any square in any of the eight directions (left, right, up, down, or the four diagonals). In the diagram below, the green circles denote all the cells the queen can attack from $(4, 4)$:



There are k obstacles on the chessboard preventing the queen from attacking any square that has an obstacle blocking the the queen's path to it. For example, an obstacle at location $(3, 5)$ in the diagram above would prevent the queen from attacking cells $(3, 5)$, $(2, 6)$, and $(1, 7)$:



Given the queen's position and the locations of all the obstacles, find and print the number of squares the queen can attack from her position at (r_q, c_q).

Input Format

The first line contains two space-separated integers describing the respective values of n (the side length of the board) and k (the number of obstacles).

The next line contains two space-separated integers describing the respective values of r_q and c_q, denoting the position of the queen.

Each line i of the k subsequent lines contains two space-separated integers describing the respective values of r_i and c_i, denoting the position of obstacle i.

Constraints

- $0 < n \leq 10^5$
- $0 \leq k \leq 10^5$
- A single cell may contain more than one obstacle; however, it is guaranteed that there will never be an obstacle at position (r_q, c_q) where the queen is located.

Subtasks

For 30% of the maximum score:

- $0 < n \leq 100$
- $0 \leq k \leq 100$

For 55% of the maximum score:

- $0 < n \leq 1000$
- $0 \leq k \leq 10^5$

Output Format

Print the number of squares that the queen can attack from position (r_q, c_q).

Sample Input 0

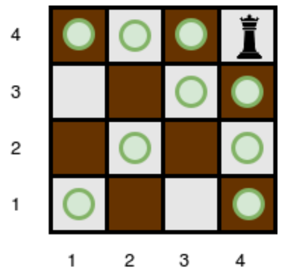
```
4 0
4 4
```

Sample Output 0

```
9
```

Explanation 0

The queen is standing at position $(4, 4)$ on a 4×4 chessboard with no obstacles:



We then print the number of squares she can attack from that position, which is 9.

Sample Input 1

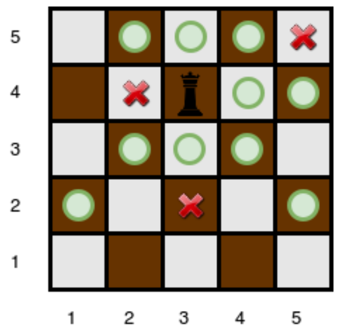
```
5 3
4 3
5 5
4 2
2 3
```

Sample Output 1

```
10
```

Explanation 1

The queen is standing at position $(4, 3)$ on a 5×5 chessboard with $k = 3$ obstacles:



We then print the number of squares she can attack from that position, which is 10.

Current Buffer (saved locally, editable) Java 7

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    static int queensAttack(int n, int k, int r_q, int c_q, int[][] 
obstacles) {
        // Complete this function
    }

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int n = in.nextInt();
        int k = in.nextInt();
        int r_q = in.nextInt();
        int c_q = in.nextInt();
        int[][] obstacles = new int[k][2];
        for(int obstacles_i = 0; obstacles_i < k; obstacles_i++){
            for(int obstacles_j = 0; obstacles_j < 2; obstacles_j++){
                obstacles[obstacles_i][obstacles_j] = in.nextInt();
            }
        }
        int result = queensAttack(n, k, r_q, c_q, obstacles);
        System.out.println(result);
        in.close();
    }
}

```

Line: 1 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature