

Problem

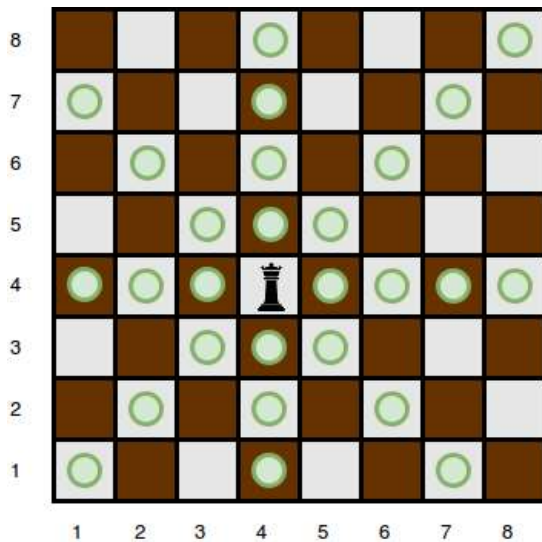
You will be given a square chess board with one queen and a number of obstacles placed on it. Determine how many squares the queen can attack.

A queen is standing on an chessboard. The chess board's rows are numbered from to , going from bottom to top. Its columns are numbered from to , going from left to right. Each square is referenced by a tuple, , describing the row, , and column, , where the square is located.

Submissions

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

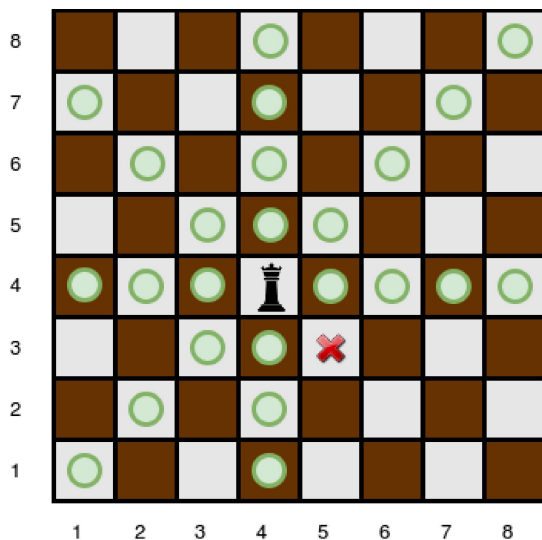
Leaderboard



Discussions

Editorial

There are obstacles on the chessboard, each preventing the queen from attacking any square beyond it on that path. For example, an obstacle at location in the diagram above prevents the queen from attacking cells , , and :



Given the queen's position and the locations of all the obstacles, find

Change Theme Language C++11

```

44  int queensAttack(int n, int r_q, int c_q, vector<vector<int>>> obstacles) {
45  }
46
47  int main() {
48      int n, k;
49      cin >> n >> k;
50
51      int r_q, c_q;
52      cin >> r_q >> c_q;
53
54      vector<vector<int>> obstacles(k, vector<int>(2));
55      for (int i = 0; i < k; i++) {
56          cin >> obstacles[i][0] >> obstacles[i][1];
57      }
58
59      cout << queensAttack(n, k, r_q, c_q, obstacles) << endl;
60
61      return 0;
62  }
63

```

Line: 7 Col: 1

Upload Code as File

☐ Test against custom input

Run Code

Submit Code

✓ Test case 0

✓ Test case 1

✓ Test case 2

✓ Test case 3

✓ Test case 4

✓ Test case 5

✓ Test case 6

Compiler Message

Success

Input (stdin)

```

1 4 0
2 4 4

```

Download

Expected Output

```

1 9

```

Download

