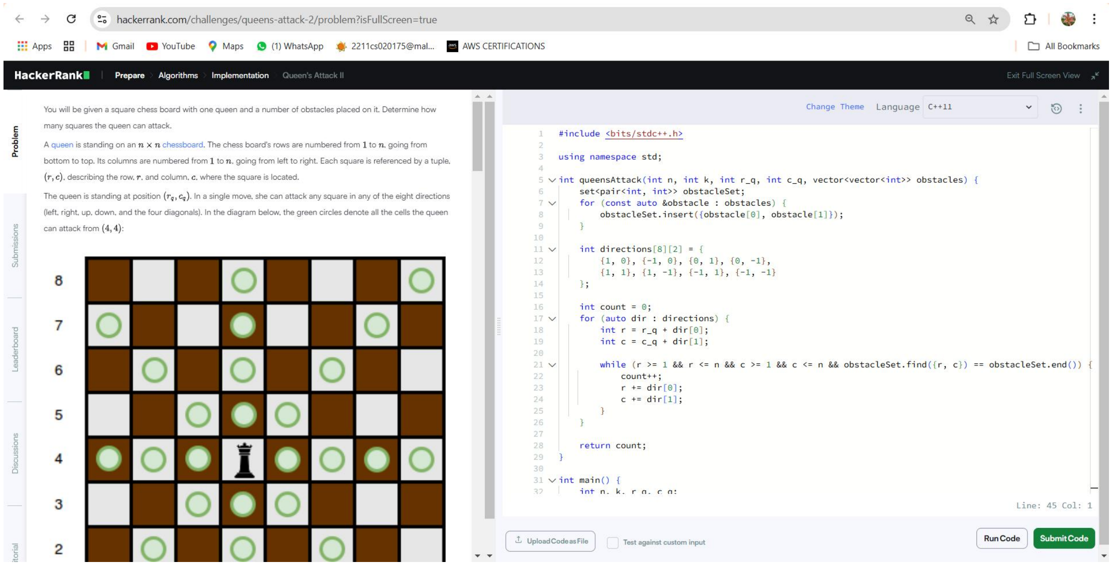
DAA ASSIGNMENT (2211CS020175)

Queens Attack - II

https://www.hackerrank.com/challenges/queens-attack-2/problem?isFullScreen=true.



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 $n \times n$ chessboard. The chess board'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 is referenced 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). 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 $(4, 4)$:

```cpp
#include <bits/stdc++.h>

using namespace std;

int queensAttack(int n, int k, int r_q, int c_q, vector<vector<int>> obstacles) {
    set<pair<int, int>> obstacleSet;
    for (const auto &obstacle : obstacles) {
        obstacleSet.insert({obstacle[0], obstacle[1]});
    }

    int directions[8][2] = {
        {1, 0}, {-1, 0}, {0, 1}, {0, -1},
        {1, 1}, {1, -1}, {-1, 1}, {-1, -1}
    };

    int count = 0;
    for (auto dir : directions) {
        int r = r_q + dir[0];
        int c = c_q + dir[1];

        while (r >= 1 && r <= n && c >= 1 && c <= n && obstacleSet.find({r, c}) == obstacleSet.end()) {
            count++;
            r += dir[0];
            c += dir[1];
        }
    }

    return count;
}

int main() {
    int n, k, r_q, c_q;
```

Done By:
2211cs020175
Aiml-Theta