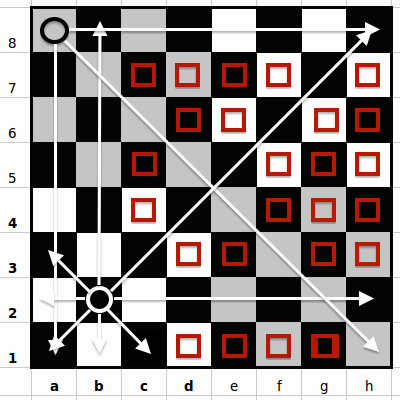
The are some queens on the chessboard. Find the number of cells which none of the queens can reach.

**Example:**

queens([["a8"],["b2"]]) = 25

Here's a reference picture to the chessboard from the example:  
  
The circles represent the queens, the arrows represent the cells the queens can visit, and the boxes mark the cells that are out of the queens' reach.

* **[input] array.string q**
  + Queens positions. Each position is given in the form "xy", where 'a' ≤ x ≤ 'h' and 1 ≤ y ≤ 8.
* **[output] integer**
  + The number of cell none of the queens can reach.

<https://codefights.com/challenge/R5JxKkLzzdg3S5RYu>

#include <iostream>

#include <stdio.h>

#include <vector>

#include <conio.h>

int queens(std::vector<std::string> q)

{

    struct Helper {

        std::vector<int> ubicarReina(std::string ub)

        {

            char fila = ub[1];

            char col = ub[0];

            int resFila = 0;

            switch (fila)

            {

                case '8':

                    resFila = 0;

                    break;

                case '7':

                    resFila = 1;

                    break;

                case '6':

                    resFila = 2;

                    break;

                case '5':

                    resFila = 3;

                    break;

                case '4':

                    resFila = 4;

                    break;

                case '3':

                    resFila = 5;

                    break;

                case '2':

                    resFila = 6;

                    break;

                case '1':

                    resFila = 7;

                    break;

            }

            int resCol = 0;

            switch (col)

            {

                case 'a':

                    resCol = 0;

                    break;

                case 'b':

                    resCol = 1;

                    break;

                case 'c':

                    resCol = 2;

                    break;

                case 'd':

                    resCol = 3;

                    break;

                case 'e':

                    resCol = 4;

                    break;

                case 'f':

                    resCol = 5;

                    break;

                case 'g':

                    resCol = 6;

                    break;

                case 'h':

                    resCol = 7;

                    break;

            }

            std::vector<int> res;

            res.push\_back(resFila);

            res.push\_back(resCol);

            return res;

        }

    };

    bool tablero[8][8];

    for(int i =0; i < 8; i++)

        for(int j =0; j<8; j++)

            tablero[i][j]=false;

    Helper h;

    for (int i = 0; i < q.size(); i++)

    {

        int f = h.ubicarReina(q[i])[0];

        int c = h.ubicarReina(q[i])[1];

        tablero[f][c] = true;

        for (int j = 0; j < 8; j++)

        {

            for (int k = 0; k < 8; k++)

            {

                if (f == j || c == k)

                {

                    tablero[j][k] = true;

                }

            }

        }

        //int j = f, k = c;

        int fila = f, col = c;

        //NE

        while (fila >= 0 && col < 8)

        {

            tablero[fila][col] = true;

            fila--;

            col++;

        }

        //SE

        fila = f;

        col = c;

        while (fila < 8 && col < 8)

        {

            tablero[fila][col] = true;

            fila++;

            col++;

        }

        //NO

        fila = f;

        col = c;

        while (fila >= 0 && col >= 0)

        {

            tablero[fila][col] = true;

            fila--;

            col--;

        }

        //SO

        fila = f;

        col = c;

        while (fila < 8 && col >= 0)

        {

            tablero[fila][col] = true;

            fila++;

            col--;

        }

    }

    int cont = 0;

    for (int i = 0; i < 8; i++)

    {

        for (int j = 0; j < 8; j++)

        {

            if (!tablero[i][j])

            {

                cont++;

            }

        }

    }

    return cont;

}

int main(){

    std::vector<std::string> reinas;

    reinas.push\_back("a8");

    reinas.push\_back("b2");

    int res =  queens(reinas);

    printf("%d ", res );

    getch();

 return 0;

}