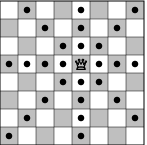
**Queen**

By Fábio Dias Moreira https://urionlinejudge.r.worldssl.net/gallery/images/flags/br.gif Brazil

**Timelimit: 1**

The game of Chess has several pieces with curious movements. One of them is the *Queen*, which can move any number of squares in any direction: in the same line, in the same column or in any of the diagonals, as illustrated by the figure below (black dots represent positions the queen may reach in one move):



The great Chess Master Kary Gasparov invented a new type of chess problem: given the position of a queen in an empty standard chess board (that is, an 8 x 8 board) how many moves are needed so that she reaches another given square in the board?

Kary found the solution for some of those problems, but is having a difficult time to solve some others, and therefore he has asked that you write a program to solve this type of problem.

**Input**

The input contains several test cases. The only line of each test case contains four integers *X1*, *Y1*, *X2* and *Y2* (*1 ≤ X1, Y1, X2, Y2 ≤ 8*). The queen starts in the square with coordinates *(X1, Y1)*, and must finish at the square with coordinates *(X2, Y2)*. In the chessboard, columns are numbered from 1 to 8, from left ro right; lines are also numbered from 1 to 8, from top to bottom. The coordinates of a square in line *X* and column *Y* are (*X, Y*).

The end of input is indicated by a line containing four zeros, separated by spaces.

**Output**

For each test case in the input your program must print a single line, containing an integer, indicating the smallest number of moves needed for the queen to reach the new position.

| **Input Sample** | **Output Sample** |
| --- | --- |
| 4 4 6 2 3 5 3 5 5 5 4 3 0 0 0 0 | 1 0 2 |

<https://www.urionlinejudge.com.br/judge/es/problems/view/1087>

--ACEPTADO---

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

while (true)

{

//string input = "4 4 6 2";

//string input = "3 5 3 5";

string input = Console.ReadLine(); // "5 5 4 3";

int X1, Y1, X2, Y2;

X1 = int.Parse(input.Split(' ')[0]);

Y1 = int.Parse(input.Split(' ')[1]);

X2 = int.Parse(input.Split(' ')[2]);

Y2 = int.Parse(input.Split(' ')[3]);

if (X1 == 0 && X2 == 0 && Y1 == 0 && Y2 == 0)

{

break;

}

int movimientos = 0;

if (X1 == X2 && Y1 == Y2)

{

Console.WriteLine("0");

continue;

//break;

}

else

{

//NORTE

int fila = X1, col = Y1;

if (col == Y2)

{

while (fila >= 0)

{

fila--;

if (fila == X2)

{

movimientos = 1;

break;

}

}

}

//NORESTE

fila = X1;

col = Y1;

while (fila >= 0 && col < 8)

{

fila--;

col++;

if (fila == X2 && col == Y2)

{

movimientos = 1;

break;

}

}

//ESTE

fila = X1;

col = Y1;

if (fila == X2)

{

while (col < 8)

{

col++;

if (col == Y2)

{

movimientos = 1;

break;

}

}

}

//SURESTE

fila = X1;

col = Y1;

while (fila < 8 && col < 8)

{

fila++;

col++;

if (fila == X2 && col == Y2)

{

movimientos = 1;

break;

}

}

//SUR

fila = X1;

col = Y1;

if (col == Y2)

{

while (fila < 8)

{

fila++;

if (fila == X2)

{

movimientos = 1;

break;

}

}

}

//SUROESTE

fila = X1;

col = Y1;

while (fila < 8 && col >= 0)

{

fila++;

col--;

if (fila == X2 && col == Y2)

{

movimientos = 1;

break;

}

}

//OESTE

fila = X1;

col = Y1;

if (fila == X2)

{

while (col >= 0)

{

col--;

if (col == Y2)

{

movimientos = 1;

break;

}

}

}

//NOROESTE

fila = X1;

col = Y1;

while (fila >= 0 && col >= 0)

{

fila--;

col--;

if (fila == X2 && col == Y2)

{

movimientos = 1;

break;

}

}

if (movimientos == 1)

{

Console.WriteLine(movimientos);

continue;

// break;

}

}

if (movimientos != 1)

{

Console.WriteLine("2");

}

}

Console.ReadLine();

}

}

}