**6 kyu**

**Chessboard Squares Under Queen's Attack**

7195% of 119 of30[curious\_db97](https://www.codewars.com/users/curious_db97)

C#

* [TRAIN AGAIN](https://www.codewars.com/kata/chessboard-squares-under-queens-attack/train/csharp)
* [NEXT KATA](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/chessboard-squares-under-queens-attack/solutions/csharp)

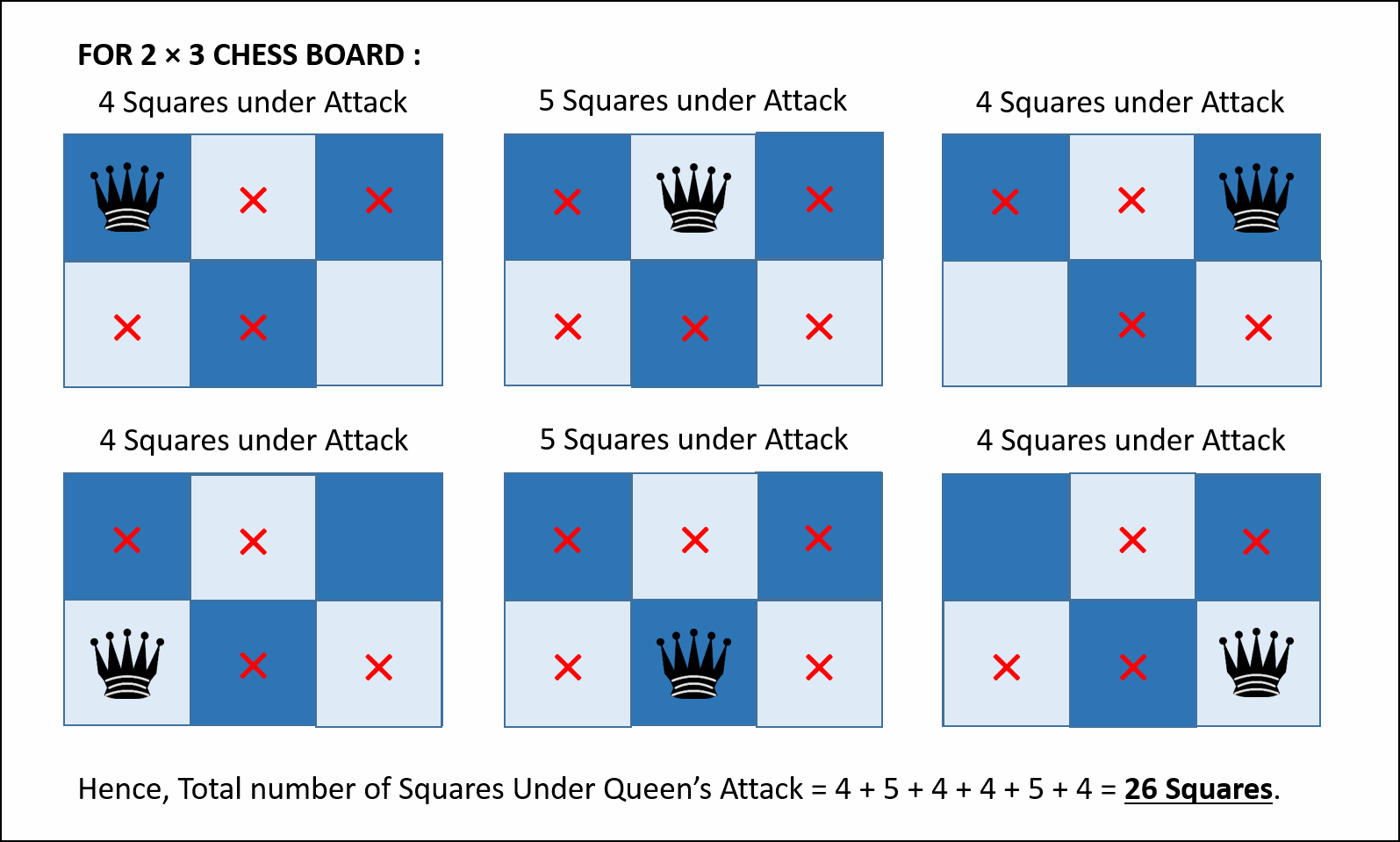
[Discourse (4)](https://www.codewars.com/kata/chessboard-squares-under-queens-attack/discuss/csharp)

* Add to Collection
* |
* Share this kata:

For a given pair of a and b : Consider a Chess board of a × b squares. Now, for each of the squares; Imagine a Queen standing on that square and compute the number of squares under the queen's attack. Add all the numbers you get for each of the a × b possible queen's position and return it.

**Examples :**

* For a = 2 and b = 2 : squaresUnderQueenAttack(2,2) => 12.
* For a = 2 and b = 3 : squaresUnderQueenAttack(2,3) => 26.

Explaination :   


**Constraints :**

* 1 ≤ a ≤ 20.
* 1 ≤ b ≤ 20.

<https://www.codewars.com/kata/chessboard-squares-under-queens-attack/csharp>

* [mattamattical](https://www.codewars.com/users/mattamattical)
* **using System;**
* **using System.Linq;**
* **namespace myjinxin {**
* **public class Kata {**
* **public int ChessboardSquaresUnderQueenAttack(int w, int h) =>**
* **Enumerable.Range(1, w).SelectMany(x =>**
* **Enumerable.Range(1, h).Select(y => {**
* **var r = w - x;**
* **var d = h - y;**
* **var l = x - 1;**
* **var u = y - 1;**
* **return w + h - 2 + Math.Min(r, d) + Math.Min(r, u) + Math.Min(l, d) + Math.Min(l, u);**
* **})).Sum();**
* **}**

**}**

* + - Best Practices1
    - Clever0
  + 0
  + [Fork](https://www.codewars.com/kumite/new?group_id=5a599672960f30210b000d24&review_id=5790a7e0e0e95484c30000ee)
  + Compare with your solution
  + [Link](https://www.codewars.com/kata/reviews/5790a7e0e0e95484c30000ee/groups/5a599672960f30210b000d24)
* [SandFox](https://www.codewars.com/users/SandFox)
* **namespace myjinxin**
* **{**
* **using System;**
* **public class Kata**
* **{**
* **public int ChessboardSquaresUnderQueenAttack(int a,int b){**
* **//coding here...**
* **int min = Math.Min(a, b);**
* **int max = Math.Max(a, b);**
* **return a\*b\*(a+b-2) + 2 \* ( 2\* (min - 1) \* min \* (min + 1)/3 + (max-min -1) \* (min - 1) \* min);**
* **}**
* **}**

**}**

* + - Best Practices0
    - Clever2
  + 0
  + [Fork](https://www.codewars.com/kumite/new?group_id=5939d9507e3d68975a000036&review_id=5790a7e0e0e95484c30000ee)
  + Compare with your solution
  + [Link](https://www.codewars.com/kata/reviews/5790a7e0e0e95484c30000ee/groups/5939d9507e3d68975a000036)
* [nachoMonllor](https://www.codewars.com/users/nachoMonllor)
* **namespace myjinxin**
* **{**
* **using System;**
* **public class Kata**
* **{**
* **public int ChessboardSquaresUnderQueenAttack(int a,int b){**
* **int cont = 0;**
* **for (int i = 1; i <= a; i++)**
* **{**
* **for (int j = 1; j <= b; j++)**
* **{**
* **//if(Math.Abs())**
* **for (int k = 1; k <= a; k++)**
* **{**
* **for (int l = 1; l <= b; l++)**
* **{**
* **if ((k == i && b != j) || (b == j && k != i))**
* **{**
* **cont++;**
* **}**
* **if (k != i && l != j && Math.Abs(k - i) == Math.Abs(l - j))**
* **{**
* **cont++;**
* **}**
* **}**
* **}**
* **}**
* **}**
* **return cont;**
* **}**
* **}**

**}**

* + - Best Practices0
    - Clever0
  + 0
  + [Fork](https://www.codewars.com/kumite/new?group_id=5bf20f341a16b08ce0000cc6&review_id=5790a7e0e0e95484c30000ee)
  + Compare with your solution
  + [Link](https://www.codewars.com/kata/reviews/5790a7e0e0e95484c30000ee/groups/5bf20f341a16b08ce0000cc6)
* [JPcheck](https://www.codewars.com/users/JPcheck)
* **namespace myjinxin**
* **{**
* **using System;**
* **public class Kata**
* **{**
* **public int ChessboardSquaresUnderQueenAttack(int a,int b){**
* **int result = 0;**
* **for(int i=1;i<=a;i++)**
* **for(int j =1;j<=b;j++)**
* **result+=Place(a,b,i,j);**
* **return result;**
* **}**
* **public int Place(int a,int b, int x, int y)**
* **{**
* **int result=a+b-2;**
* **//diags**
* **int i=1;**
* **//UL**
* **while(x-i>0 && y-i>0)**
* **{result+=1;i++;}**
* **//UR**
* **i=1;**
* **while(x-i>0 && y+i<=b)**
* **{result+=1;i++;}**
* **//DL**
* **i=1;**
* **while(x+i<=a && y-i>0)**
* **{result+=1;i++;}**
* **//DR**
* **i=1;**
* **while(x-i>0 && y+i<=b)**
* **{result+=1;i++;}**
* **return result;**
* **}**
* **}**

**}**

* + - Best Practices0
    - Clever0
  + 0
  + [Fork](https://www.codewars.com/kumite/new?group_id=5adf58eb60da5c618b003903&review_id=5790a7e0e0e95484c30000ee)
  + Compare with your solution
  + [Link](https://www.codewars.com/kata/reviews/5790a7e0e0e95484c30000ee/groups/5adf58eb60da5c618b003903)
* [Sasha Markov](https://www.codewars.com/users/Sasha%20Markov)
* **namespace myjinxin {**
* **public class Kata {**
* **public int ChessboardSquaresUnderQueenAttack( int a, int b ) {**
* **var result = 0;**
* **for ( int i = 0; i < a; i++ ) {**
* **for ( int j = 0; j < b; j++ ) {**
* **var t = 0;**
* **t += ( a - 1 );**
* **t += ( b - 1 );**
* **var x = i + 1;**
* **var y = j + 1;**
* **while ( x < a && y < b ) {**
* **t += 1;**
* **x += 1;**
* **y += 1;**
* **}**
* **x = i + 1;**
* **y = j - 1;**
* **while ( x < a && y >= 0 ) {**
* **t += 1;**
* **x += 1;**
* **y -= 1;**
* **}**
* **x = i - 1;**
* **y = j + 1;**
* **while ( x >= 0 && y < b ) {**
* **t += 1;**
* **x -= 1;**
* **y += 1;**
* **}**
* **x = i - 1;**
* **y = j - 1;**
* **while ( x >= 0 && y >= 0 ) {**
* **t += 1;**
* **x -= 1;**
* **y -= 1;**
* **}**
* **result += t;**
* **}**
* **}**
* **return result;**
* **}**
* **}**

**}**

* + - Best Practices0
    - Clever0
  + 0
  + [Fork](https://www.codewars.com/kumite/new?group_id=5a72033c0202bc8970000259&review_id=5790a7e0e0e95484c30000ee)
  + Compare with your solution
  + [Link](https://www.codewars.com/kata/reviews/5790a7e0e0e95484c30000ee/groups/5a72033c0202bc8970000259)
* [hardgib](https://www.codewars.com/users/hardgib)
* **using System;**
* **namespace myjinxin**
* **{**
* **using System;**
* **public class Kata**
* **{**
* **public int ChessboardSquaresUnderQueenAttack(int a, int b)**
* **{**
* **int result = 0;**
* **for (int i = 0; i < a; i++) // горизонталь**
* **for (int j = 0; j < b; j++) // вертикаль**
* **{**
* **for (int k = 0; k < a; k++) // подсчет по горизонтали**
* **if (k != i) result++;**
* **for (int k = 0; k < b; k++) // подсчет по вертикали**
* **if (k != j ) result++;**
* **int m = i; int n = j; // подсчет вниз-вправо**
* **while (m+1 < a && n+1 < b)**
* **{**
* **result++;**
* **m++; n++;**
* **}**
* **m = i; n = j; // подсчет влево-вверх**
* **while (m > 0 && n > 0)**
* **{**
* **result++;**
* **m--; n--;**
* **}**
* **m = i; n = j; // подсчет вправо-вверх**
* **while (m + 1 < a && n > 0)**
* **{**
* **result++;**
* **m++; n--;**
* **}**
* **m = i; // подсчет влево-вниз**
* **n = j;**
* **while (m > 0 && n + 1 < b)**
* **{**
* **result++;**
* **m--; n++;**
* **}**
* **}**
* **return result;**
* **}**
* **}**

**}**

* + - Best Practices0
    - Clever0
  + 0
  + [Fork](https://www.codewars.com/kumite/new?group_id=5a54e7a56f46dec1e5000ac0&review_id=5790a7e0e0e95484c30000ee)
  + Compare with your solution
  + [Link](https://www.codewars.com/kata/reviews/5790a7e0e0e95484c30000ee/groups/5a54e7a56f46dec1e5000ac0)
* [KataSideKick](https://www.codewars.com/users/KataSideKick)
* **namespace myjinxin**
* **{**
* **using System;**
* **public class Kata**
* **{**
* **public int ChessboardSquaresUnderQueenAttack(int a, int b)**
* **{**
* **var sum = 0;**
* **for (int i = 0; i < a; i++)**
* **{**
* **for (int j = 0; j < b; j++)**
* **{**
* **sum += a + b - 2;**
* **sum += Math.Min(i, j);**
* **sum += Math.Min(a - i - 1, j);**
* **sum += Math.Min(i, b - j - 1);**
* **sum += Math.Min(a - i - 1, b - j - 1);**
* **}**
* **}**
* **return sum;**
* **}**
* **}**

**}**

* + - Best Practices0
    - Clever0
  + 0
  + [Fork](https://www.codewars.com/kumite/new?group_id=5a53bfb8dd0fd22d3b000004&review_id=5790a7e0e0e95484c30000ee)
  + Compare with your solution
  + [Link](https://www.codewars.com/kata/reviews/5790a7e0e0e95484c30000ee/groups/5a53bfb8dd0fd22d3b000004)
* [WhoIsMichael](https://www.codewars.com/users/WhoIsMichael)
* **namespace myjinxin**
* **{**
* **using System;**
* **public class Kata**
* **{**
* **public int ChessboardSquaresUnderQueenAttack(int a,int b)**
* **{**
* **int[,] matr = new int[a, b];**
* **int count = 0;**
* **for(int i = 0; i < a; i++)**
* **for(int j = 0; j < b; j++)**
* **{**
* **for (int k = i + 1; k < a; k++)**
* **count++;**
* **for (int k = i - 1; k >= 0; k--)**
* **count++;**
* **for (int k = j + 1; k < b; k++)**
* **count++;**
* **for (int k = j - 1; k >= 0; k--)**
* **count++;**
* **for (int k = i + 1, t = j + 1; k < a && t < b; k++, t++)**
* **count++;**
* **for (int k = i - 1, t = j - 1; k >= 0 && t >= 0; k--, t--)**
* **count++;**
* **for (int k = i + 1, t = j - 1; k < a && t >= 0; k++, t--)**
* **count++;**
* **for (int k = i - 1, t = j + 1; k >= 0 && t < b; k--, t++)**
* **count++;**
* **}**
* **return count;**
* **}**
* **}**

**}**

* + - Best Practices0
    - Clever0
  + 0
  + [Fork](https://www.codewars.com/kumite/new?group_id=5a50d25a0817cf24420002d9&review_id=5790a7e0e0e95484c30000ee)
  + Compare with your solution
  + [Link](https://www.codewars.com/kata/reviews/5790a7e0e0e95484c30000ee/groups/5a50d25a0817cf24420002d9)
* [myjinxin2015](https://www.codewars.com/users/myjinxin2015)
* **namespace myjinxin**
* **{**
* **using System;**
* **public class Kata**
* **{**
* **public int ChessboardSquaresUnderQueenAttack(int a,int b){**
* **var rs=0;**
* **for (int i=0;i<a;i++) for (int j=0;j<b;j++) rs+=GetN(a,b,i,j); return rs;**
* **}**
* **public int GetN(int a,int b,int i,int j){**
* **var n=0;**
* **n+=a+b-2;**
* **for(int ii=i-1,jj=j-1;ii>-1&&jj>-1;ii--,jj--) n++;**
* **for(int ii=i-1,jj=j+1;ii>-1&&jj<b; ii--,jj++) n++;**
* **for(int ii=i+1,jj=j-1;ii<a&&jj>-1; ii++,jj--) n++;**
* **for(int ii=i+1,jj=j+1;ii<a&&jj<b; ii++,jj++) n++;**
* **return n;**
* **}**
* **}**

**}**

* + - Best Practices0
    - Clever0
  + 0
  + [Fork](https://www.codewars.com/kumite/new?group_id=5790b09b39f33a5359000015&review_id=5790a7e0e0e95484c30000ee)
  + Compare with your solution
  + [Link](https://www.codewars.com/kata/reviews/5790a7e0e0e95484c30000ee/groups/5790b09b39f33a5359000015)
* [myjinxin2015](https://www.codewars.com/users/myjinxin2015)
* **namespace myjinxin**
* **{**
* **using System;**
* **//using System.Linq;**
* **//using System.Text.RegularExpressions;**
* **public class Kata**
* **{**
* **public int ChessboardSquaresUnderQueenAttack(int a,int b){**
* **var rs=0;**
* **for (int i=0;i<a;i++) for (int j=0;j<b;j++) rs+=GetN(a,b,i,j); return rs;**
* **}**
* **public int GetN(int a,int b,int i,int j){**
* **var n=0;**
* **n+=a+b-2;**
* **for(int ii=i-1,jj=j-1;ii>-1&&jj>-1;ii--,jj--) n++;**
* **for(int ii=i-1,jj=j+1;ii>-1&&jj<b; ii--,jj++) n++;**
* **for(int ii=i+1,jj=j-1;ii<a&&jj>-1; ii++,jj--) n++;**
* **for(int ii=i+1,jj=j+1;ii<a&&jj<b; ii++,jj++) n++;**
* **return n;**
* **}**
* **}**

**}**