**7 kyu**

**Principal Diagonal | VS | Secondary Diagonal**

6290% of 8756 of327[CRYPTON](https://www.codewars.com/users/CRYPTON)

C++

* [TRAIN AGAIN](https://www.codewars.com/kata/principal-diagonal-vs-secondary-diagonal/train/cpp)
* [NEXT KATA](https://www.codewars.com/trainer/cpp)

Details

[Solutions](https://www.codewars.com/kata/principal-diagonal-vs-secondary-diagonal/solutions/cpp)

[Discourse (17)](https://www.codewars.com/kata/principal-diagonal-vs-secondary-diagonal/discuss/cpp)

* Add to Collection
* |
* Share this kata:

Principal Diagonal <--| VS |--> Secondary Diagonal

W-A-R

Principal Diagonal -- The principal diagonal in a matrix identifies those elements of the matrix running from north-west to south-east. An example of this diagonal :

[1,0,0]  
[0,1,0]  
[0,0,1]

elements of Principal Diagonal 1, 1, 1 .

Secondary Diagonal -- the secondary diagonal of amatrix identifies those elements of the matrix running from north-east to south-west. An example of this diagonal :

[0,0,1]  
[0,5,0]  
[2,0,0]

elements of Seconrady Diagonal 1, 5, 2 .



3+8+2 > 0+8+0 => return 'Principal diagonal win!'

Your task is to find which diagonal is larger (which a diagonal have bigger sum of their elements).

If the primary diagonal is larger,--> return "Principal Diagonal win!".

If the secondary diagonal is larger,--> return "Secondary Diagonal win!".

if it's draw,--> return "Draw!".

You will always receive matrices of the same dimension

This is my first kata, if you like it, please rate it :D

<https://www.codewars.com/kata/principal-diagonal-vs-secondary-diagonal/cpp>

[lszcs](https://www.codewars.com/users/lszcs), [990022](https://www.codewars.com/users/990022)

**#include <string>**

**#include <vector>**

**std::string diagonal(std::vector<std::vector<int>> matrix) {**

**long long sum = 0;**

**for (unsigned int i = 0; i < matrix.size(); ++i)**

**{**

**sum += matrix[i][i];**

**sum -= matrix[i][matrix.size() - i - 1];**

**}**

**return sum==0 ? "Draw!" : (sum>0 ? "Principal Diagonal win!" : "Secondary Diagonal win!");**

**};**

#include <string>

#include <vector>

using namespace std;

std::string diagonal(vector<vector<int> > matrix){

  //your code here

    int n = matrix.size();

    int f =0, c = n-1;

    int sp =0, ss =0;

    for(int i =0; i<n; i++) {

        sp += matrix[i][i];

        ss += matrix[f][c];

        f++;

        c--;

    }

    if(sp > ss) return "Principal diagonal win!";

    else if( ss > sp) return "Secondary Diagonal win!";

    return "Draw!";

}