**EJERCICIO 19**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Challenge "Menu"**

In this exercise we will to create a menu for our "Chess Game".

The menu needs to have this structure and the player/players can navigate in it until they decide to exit the game from the main menu.

* Game
  + Start a new game
    - **It must show an interactive menu that asks for the names of each user, (if the user does not exist, it offers to create it) and assigns a color for the game.**
  + Load a started game
    - **It must show all the games started but not finished. Use dummy information.**
  + Play a finished game
    - **You must show all completed games (Use dummy data) and offer to replay it.**
  + Back to menu
* Players
  + Create a new player
    - **It must show an interactive menu that asks for user information.**
  + List all players
  + List players score
  + Update player
    - **You must show all players (Use dummy data) and offer to update it by an interactive menu.**
  + Delete player
    - **You must show all players (Use dummy data) and offer to delete it by an interactive menu.**
  + Back to menu
* Help
  + How to play
  + Game rules
  + Contact Admin
    - **You must show an interactive menu to insert email (of the player) and insert a message to be send.**
  + Back to menu
* Exit

**Input Format**

*Feel free...*

Define what will be the command entry strategy to operate the menu

**Output Format**

The menu with all requirements and error handling (for example, incorrect user input)

**CHALLENGE QUEEN**

[![Review Assignment Due Date](https://classroom.github.com/assets/deadline-readme-button-24ddc0f5d75046c5622901739e7c5dd533143b0c8e959d652212380cedb1ea36.svg)](https://classroom.github.com/a/0jXkpPUb)  
# Challenge "Queen's Movements"  
  
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](https://en.wikipedia.org/wiki/Queen\_%28chess%29) is standing on   
an **\*\****\_*n x n*\_***\*\*** [chessboard](https://en.wikipedia.org/wiki/Chess).   
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<sub>q</sub>, c<sub>q</sub>)*\_***\*\***.   
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)*\_***\*\***:  
  
![img](/docs/\_images/img.png)  
  
There are obstacles on the chessboard, each preventing the queen from   
attacking any square beyond it on that path. For example, an obstacle at   
location **\*\****\_*(3, 5)*\_***\*\*** in the diagram above prevents the queen from attacking   
cells **\*\****\_*(3, 5)*\_***\*\***, **\*\****\_*(2, 6)*\_***\*\*** and **\*\****\_*(1, 7)*\_***\*\***:  
  
![img](/docs/\_images/img\_1.png)  
  
Given the queen's position and the locations of all the obstacles, find and   
print the number of squares the queen can attack from her position   
at **\*\****\_*(r<sub>q</sub>, c<sub>q</sub>)*\_***\*\***. In the board above,   
there are **\*\****\_*24*\_***\*\*** such squares.  
  
## Funtion Description  
  
Complete the queensAttack function in the editor below.  
  
queensAttack has the following parameters:  
*-* int **\*\****\_*n*\_***\*\***: the number of rows and columns in the board  
*-* nt **\*\****\_*k*\_***\*\***: the number of obstacles on the board  
*-* int **\*\****\_*r\_q*\_***\*\***: the row number of the queen's position  
*-* int **\*\****\_*c\_q*\_***\*\***: the column number of the queen's position  
*-* int obstacles **\*\****\_*[k][2]*\_***\*\***: each element is an array of **\*\****\_*2*\_***\*\*** integers, the row and column of an obstacle  
  
## Returns  
  
*-* int: the number of squares the queen can attack  
  
## Input Format  
  
The first line contains two space-separated integers **\*\****\_*n*\_***\*\*** and **\*\****\_*k*\_***\*\***,   
the length of the board's sides and the number of obstacles.  
The next line contains two space-separated integers **\*\****\_*r<sub>q</sub>*\_***\*\*** and **\*\****\_*c<sub>q</sub>*\_***\*\***,   
the queen's row and column position.  
Each of the next **\*\****\_*k*\_***\*\*** lines contains two space-separated integers **\*\****\_*r[i]*\_***\*\*** and **\*\****\_*c[i]*\_***\*\***,   
the row and column position of **\*\****\_*obstacle[i]*\_***\*\***.  
  
## Constrains  
  
![img](/docs/\_images/img\_2.png)  
  
## Subtasks  
  
![img](/docs/\_images/img\_3.png)  
  
## Sample Input 0  
  
```  
Welcome to Queens Attack!  
Chessboard size:  
4  
Number of obstacles:  
0  
Queen start position:  
Row:  
4  
Column:  
4  
```  
  
## Sample Output 0  
  
```  
Result:  
9  
```  
  
## Explanation 0  
  
The queen is standing at position **\*\****\_*(4, 4)*\_***\*\*** on a **\*\****\_*4 x 4*\_***\*\*** chessboard with no obstacles:  
  
![img](/docs/\_images/img\_4.png)  
  
The number of squares she can attack from that position is **\*\****\_*9*\_***\*\***.  
  
## Sample Input 1  
  
```  
Welcome to Queen's Attack!  
Chessboard size:  
5  
Number of obstacles:  
3  
Queen start position:  
Row:  
4  
Column:  
3  
Obstacle 1:  
Row:  
5  
Column:  
5  
Obstacle 2:  
Row:  
4  
Column:  
2  
Obstacle 3:  
Row:  
2  
Column:  
3  
```  
  
## Sample Output 1  
  
```  
Result:  
10  
```  
  
## Explanation 1  
  
The queen is standing at position **\*\****\_*(4, 3)*\_***\*\*** on a **\*\****\_*5 x 5*\_***\*\*** chessboard with **\*\****\_*k = 3*\_***\*\*** obstacles:  
  
![img](/docs/\_images/img\_5.png)  
  
The number of squares she can attack from that position is **\*\****\_*10*\_***\*\***.  
  
## Sample Input 2  
  
```  
Welcome to Queen's Attack!  
Chessboard size:  
1  
Number of obstacles:  
0  
Queen start position:  
Row:  
1  
Column:  
1  
```  
  
## Sample Output 2  
  
```  
Result:  
0  
```  
  
## Explanation 2  
  
Since there is only one square, and the queen is on it, the queen can move **\*\****\_*0*\_***\*\*** squares.