- Porque esse exercício foi solicitado para vocês?

  R: Creio que seja para saber em qual nível de conhecimento em C e em lógia a sala se encontra, para que possamos tentar implementar as teorias apresentadas em sala, e para que com o decorrer das aulas possamos pelo menos enxergar como tornar este código mais performático.

- Nome Completo: Rafael Vilas Boas de Carvalho
- RA: 215214117

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
   int squareSize = 7; //sets the square size
   int line;
   int column;
   int square[squareSize][squareSize];
   int lineResult = 0;
   int columnResult = 0;
   int mainDiagonal = 0;
   int secondaryDiagonal = 0;
   int i = squareSize-1;
   int previousValue;
   int perfectSquare = 0;

   printf("Magic Square test!\nInsert the numbers that you want to populate the square with (left to
right)\n Square Size: %d x %d\n", squareSize, squareSize);

 //populates the array
   for(line = 0; line < squareSize; line++){
      for(column = 0; column < squareSize; column++){
         scanf("%d", &square[line][column]);
      }
   }

   //checks if there are repeating numbers
   for(line = 0; line < squareSize; line++){
      for(column = 0; column < squareSize; column++){
         previousValue = square[line][column];
         if(line != 0){
            for(int z = 0; z < squareSize; z++){
               for(int y = 0; y < squareSize; y++){
                  if (line != z && column != y && previousValue == square[z][y]){
                     printf("You cannot repeat the numbers!");
                     return 0;
                  }
               }
            }
         }
```

```
   }
}

//calculates and shows the line sum
for(line = 0; line < squareSize; line++){
   for(column = 0; column < squareSize; column++){
      lineResult = lineResult + square[line][column];
   }

   if(line == 0){
         previousValue = lineResult;
   }else if (previousValue == lineResult){
         perfectSquare = 1;
   }
    printf(" Sum line %d = %d\n", (line+1), lineResult);
    lineResult = 0;
}

//calculates and shows the column sum
for(column = 0; column < squareSize; column++){
   for(line = 0; line < squareSize; line++){
      columnResult = columnResult + square[line][column];
   }
   if(column == 0){
         previousValue = columnResult;
   }else if (previousValue == columnResult){
         perfectSquare = 1;
   }
    printf(" Sum column %d = %d\n", (column+1), columnResult);
    columnResult = 0;
}

//calculates and show the main diagonal sum
for(line = 0; line < squareSize; line++){
   for(column = 0; column < squareSize; column++){
      if(line == column)
      mainDiagonal = mainDiagonal + square[line][column];
   }
}
printf(" Sum main diagonal = %d\n", mainDiagonal);

//calculates and shows the secondary diagonal sum
 for(line = 0; line < squareSize; line++){
   for(column = 0; column < squareSize; column++){
      if(column == i){
         secondaryDiagonal = secondaryDiagonal + square[line][column];
         i--;
      }
   }
}
printf(" Sum scondary diagonal = %d\n", secondaryDiagonal);
//prints the array
for(line = 0; line < squareSize; line++){
```

```
      for(column = 0; column < squareSize; column++){
         if(column < squareSize-1 )  {
            printf("    %d", square[line][column]);
         }else{
            printf("    %d\n", square[line][column]);
         }
      }
   }

   //prints if the square is perfect or not
   if(perfectSquare == 1 && mainDiagonal == secondaryDiagonal){
      printf("\nThe square is magic!\n");
   }else{
      printf("\nThe square ins`t magic :( \nThe sum of the numbers in the x and y axis, main and
secondary diagonal, have diferent results!\n ");
   }
   return 0;
}
```