

Hello! My name is Lamees E from Doha College, and this is my submission to the NRich Connecting Averages problem. Here, I took my original idea of solving the problem by setting simultaneous equations with their mean, and coded a program that would find the number in the box in any 1 x 1 or 1 x 2 box.

I ask for the user's input on the numbers in the box, form a set of simultaneous equations and allow python's SymPy program to solve for x and y, which are my variables for the boxes in the problem.

I manually installed SymPy on my Thonny program. Because you cannot easily export Thonny files, I tried to create an accessible link via Trinket, so if you're curious, you can try my code out using this link:

<https://trinket.io/python3/3ecfae1a8a38>

I successfully tested my code out on multiple examples given by the simulation on the NRich website. It was really fun to play with a fusion of mathematics and coding, and I hope you enjoy it as much as I did! :D

Thank you so much. Here is a copy of my code, but remember, it must have SymPy math imported in order to work and solve the simultaneous equations.

My Code

```
#NRich Average Calculator
```

```
from sympy import symbols, Eq, solve
```

```
x, y = symbols('x ,y')
```

```
#Ask Grid Size
```

```
print('We are going to ask you for the grid size. Make it equal to or less than 1 x 2')
```

```
length = int(input('What is the length of your grid? '))
```

```
width = int(input('What is the width of your grid? '))
```

```
#Simultaneous Equations
```

```
if length*width > 2 or length*width < 0:
```

```
    print('Sorry, grid size is not compatible with this code.')
```

```
elif length*width == 1:
```

```
    print('We are going to ask for the number around your grid, in an anticlockwise order')
```

```
    num1 = int(input('1: '))
```

```
    num2 = int(input('2: '))
```

```
    num3 = int(input('3: '))
```

```
    num4 = int(input('4: '))
```

```
    numerator = num1 + num2 + num3 + num4
```

```
    mean = numerator/4
```

```
    print('The missing number is', mean)
```

else:

print('We are going to ask for the number around your grid, starting on top left, in anticlockwise order. If none, write 0.')

num1 = int(input('1: '))

num2 = int(input('2: '))

num3 = int(input('3: '))

num4 = int(input('4: '))

num5 = int(input('5: '))

num6 = int(input('6: '))

eq1 = Eq(y + num1 + num2 + num3, 4*x)

eq2 = Eq(x + num6 + num5 + num4, 4*y)

solution = solve((eq1, eq2), (x, y))

print('The first box is', solution[x], 'and the second box is', solution[y])