**7 kyu**

**Cantor's Diagonals**

3094% of43125[marbiru](https://www.codewars.com/users/marbiru)

Python

* [TRAIN AGAIN](https://www.codewars.com/kata/cantors-diagonals/train/python)
* [NEXT KATA](https://www.codewars.com/trainer/python)

Details

[Solutions](https://www.codewars.com/kata/cantors-diagonals/solutions/python)

[Discourse](https://www.codewars.com/kata/cantors-diagonals/discuss/python)

* Add to Collection
* |
* Share this kata:

Given a list of lists containing only 1s and 0s, return a new list that differs from list 1 in its first element, list 2 in its second element, list 3 in its 3rd element, and so on.

cantor([[0,0,0],

[1,1,1],

[0,1,0]]) = [1,0,1]

cantor([[1,0,0],

[0,1,0],

[0,0,1]]) = [0,0,0]

The nested list will always be perfectly square. Your solution should be a list containing only 1s and 0s.

See [Wikipedia](https://en.wikipedia.org/wiki/Cantor%27s_diagonal_argument) for background (if you're interested; it won't help you solve the kata). Obviously this kata is not the same because the lists are not infinite so it doesn't really prove anything -- consider it a tribute...

<https://www.codewars.com/kata/cantors-diagonals/python>

1. **def** cantor(nested\_list):
3. col = 0
4. ans = []
5. **for** i **in** range(0, len(nested\_list)):
6. **if**(nested\_list[i][col] == 1):
7. ans.append(0)
8. **else**:
9. ans.append(1)
10. col += 1
11. **return** ans
13. e = [[1,0,0],
14. [0,1,0],
15. [0,0,1]]
17. **print**( cantor(e) )