**5 kyu**

**Shuffle It Up**

9395% of 2999 of112[raulbc777](https://www.codewars.com/users/raulbc777)

Python

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

Details

[Solutions](https://www.codewars.com/kata/shuffle-it-up/solutions/python)

[Forks (3)](https://www.codewars.com/kata/shuffle-it-up/forks/python)

[Discourse (8)](https://www.codewars.com/kata/shuffle-it-up/discuss/python)

* Add to Collection
* |
* Share this kata:

We have an array of unique elements. A special kind of permutation is the one that has all of its elements in a different position than the original.

Let's see how many of these permutations may be generated from an array of four elements. We put the original array with square brackets and the wanted permutations with parentheses.

arr = [1, 2, 3, 4]

(2, 1, 4, 3)

(2, 3, 4, 1)

(2, 4, 1, 3)

(3, 1, 4, 2)

(3, 4, 1, 2)

(3, 4, 2, 1)

(4, 1, 2, 3)

(4, 3, 1, 2)

(4, 3, 2, 1)

\_\_\_\_\_\_\_\_\_\_\_\_\_

A total of 9 permutations with all their elements in different positions than arr

The task for this kata would be to create a code to count all these permutations for an array of certain length.

Features of the random tests:

l = length of the array

10 ≤ l ≤ 5000

See the example tests.

Enjoy it!

<https://www.codewars.com/kata/shuffle-it-up/python>

1. **def** all\_permuted(array\_length):
3. *# Create an array to store*
4. *# counts for subproblems*
5. *#der = [0 for i in range(array\_length + 1)]*
6. der = []
7. **for** i **in** range(array\_length + 1):
8. der.append(0)
10. *# Base cases*
11. der[0] = 1
12. der[1] = 0
13. der[2] = 1
15. *# Fill der[0..n] in bottom up manner*
16. *# using above recursive formula*
17. **for** i **in** range(3, array\_length + 1):
18. der[i] = (i - 1) \* (der[i - 1] +
19. der[i - 2])
21. *# Return result for n*
22. **return** der[array\_length]
24. **print**(all\_permuted(500))

------------------------GEEKSFORGEEKS----------------------

<https://www.geeksforgeeks.org/count-derangements-permutation-such-that-no-element-appears-in-its-original-position/>

------------------ROSETTACODE ----------------------

<https://en.wikipedia.org/wiki/Derangement>