

CSE 101 - IP
Lab 7 - Ungraded

Faro shuffle, is a shuffle in which a deck of $2n$ cards is divided into two halves. A faro shuffle that leaves the original top card at the top and the original bottom card at the bottom is known as an out-shuffle; one that moves the original top card to second and the original bottom card to second from the bottom is known as an in-shuffle. Using an in-shuffle, a deck originally arranged as 1 2 3 4 5 6 7 8 would become 5 1 6 2 7 3 8 4. Using an out-shuffle, the deck order would become 1 5 2 6 3 7 4 8.

[Source: <http://mathworld.wolfram.com/RiffleShuffle.html>]

Define a function, *shuffle(mylist)*, to perform out-shuffle on a list of n numbers (representing a deck of cards) and return the shuffled list. Here n is an even natural number.

Make use of this defined function to define another function, *howmany_shuffles(mylist)*, that takes a given list of n distinct numbers (representing a deck of cards), where n is an even natural number. This function returns the number of shuffles after which the deck would return to original order.

You should also define at least **3 test cases** for each of the two functions. The correctness of the function based on these test cases and the quality of test cases is important. Design test cases that are extensive in nature.

Sample:

`shuffle([1,2,3,4,5,6,7,8])` returns `[1,5,2,6,3,7,4,8]`

`howmany_shuffles([1,2,3,4,5,6,7,8])` returns 3. Explanation: After 3 shuffles the list would result back in `[1,2,3,4,5,6,7,8]`

You may refer to the documentation: <https://docs.python.org/3/tutorial/datastructures.html>