How Many Shuffles?

Source: Edabit
Published by **CoffeeSpoons** in **JavaScript**
Arrays, loops, math

An out-shuffle, also known as an out Faro shuffle or a perfect shuffle, is a controlled method for shuffling playing cards.

It is performed by **splitting the deck into two equal halves** and interleaving them together perfectly, with the condition that **the top card of the deck remains in place**.

Demonstration

Using an array to represent a deck of cards, an out-shuffle looks like:

```
[1, 2, 3, 4, 5, 6, 7, 8] → [1, 5, 2, 6, 3, 7, 4, 8]
// Card 1 remains in the first position.
```

If we repeat the process, the deck eventually returns to original order:

Shuffle 1:

```
[1, 2, 3, 4, 5, 6, 7, 8] → [1, 5, 2, 6, 3, 7, 4, 8]
```

Shuffle 2:

```
[1, 5, 2, 6, 3, 7, 4, 8] → [1, 3, 5, 7, 2, 4, 6, 8]
```

Shuffle 3:

```
[1, 3, 5, 7, 2, 4, 6, 8] → [1, 2, 3, 4, 5, 6, 7, 8]
// Back where we started.
```

Write a function `shuffleCount` that takes a positive even integer `num` representing the number of the cards in a deck, and returns the number of out-shuffles required to return the deck to its original order.

Examples

```
shuffleCount(8) → 3

shuffleCount(14) → 12

shuffleCount(52) → 8
```

Notes

The number of cards is always greater than zero. Thus, the smallest possible deck size is 2.