

# Extended version of Riffle Shuffles

## Appendix A: pseudocode and Python Codes

### A.1: Define a Function shuffle(deck):

```
Half1 = [top half of deck]
Half2 = [bottom half of deck]
shuffled_deck = []
Iterate x from {0} to {(length of deck) ÷ 2 - 1}:
    Add(item x of Half1) to shuffled_deck
    Add(item x of Half2) to shuffled_deck
return shuffled_deck
```

carddeck = deck of cards with (8) cards in each suit

output: carddeck

original\_deck = carddeck

carddeck = shuffle(carddeck)

output: carddeck

As long as original\_deck != carddeck:

```
carddeck = shuffle(carddeck)
```

output: carddeck

The purpose of the function shuffle(d) is to simulate a riffle shuffle on a deck of cards d, which is in the form of a list. As this function actually shuffles the deck, it does not consider the number of cards. Hence, it will work regardless of whether or not the deck is a power of 2.

#Define a function to shuffle a given deck of cards

```
def shuffle(d):
    d1 = d[:len(d)//2]
    d2 = d[len(d)//2:]
    fd = []
    for a in zip(d1, d2):
        fd += list(a)
    return fd
```

#Code to create a deck

```
deck = []
for i in ['s', 'c', 'd', 'h']:
    deck += [str(x) + i for x in range(1, 5)]
```

This number is the number of cards in each suit plus one, and can be altered as per the need.

#This last part of the program shuffles the deck until it has come back to the original order.

```
print(deck)
ordeck = deck
deck = shuffle(deck)
print(deck)
while ordeck != deck:
    deck = shuffle(deck)
print(deck)
```

A.2: This program is the same as A.1, only with a small tweak that displays the number of slots between the cards 1s and 2s for every shuffle (the changes were applied only to the last part of the program that prints each shuffle).

```
#Define a function to shuffle a given deck of cards
```

```
def shuffle(d):  
    d1 = d[:len(d)//2]  
    d2 = d[len(d)//2:]  
    fd = []  
    for a in zip(d1, d2):  
        fd += list(a)  
    if len(d2) % 2 == 1: fd.append(d2[-1])  
    return fd
```

```
#Code to create a deck
```

```
deck = []  
for i in ['s', 'c', 'd', 'h']:  
    deck += [str(x) + i for x in range(1, 5)]
```

```
#This last part of the program shuffles the deck until it has come back to the original order.
```

```
print(deck)  
print("2s is 1 slot away.")  
ordeck = deck  
deck = shuffle(deck)  
print(deck)  
print("2s is", deck.index('2s'), "slots away.")  
while ordeck != deck:  
    deck = shuffle(deck)  
    print(deck)  
    if deck.index('2s') == 1:  
        print("2s is 1 slot away.")  
    else:  
        print("2s is", deck.index('2s'), "slots away.")
```

A.3: This program is very similar to A.1, except the cards are in the form of plain integers. They are also separated by a colon, dividing the cards held in the right hand and the cards held in the left hand. In order to be displayed in this manner, an additional function, display(l) is used.

**#Define a function to shuffle a given deck of cards**

```
def shuffle(d):
    d1 = d[:len(d)//2]
    d2 = d[len(d)//2:]
    fd = []
    for a in zip(d1, d2):
        fd += list(a)
    if len(d2) % 2 == 1: fd.append(d2[-1])
    return fd
def display(l):
    for x in l[:len(l)//2]:
        print(x, end = " ")
    print(":", end = " ")
    for x in l[len(l)//2:]:
        print(x, end = " ")
    print("")
```

**#Code to create a deck**

```
deck = [i for i in range(1, 25)]
```

**#This last part of the program shuffles the deck until it has come back to the original order.**

```
display(deck)
ordeck = deck
deck = shuffle(deck)
display(deck)
while ordeck != deck:
    deck = shuffle(deck)
    display(deck)
```

A.4: This program inputs the number of cards in a deck and outputs the number of shuffles needed to restore its original order, along with the value of k. n is calculated using the formula derived in Section 5 and the while loop iterates through integer values of k and calculates the corresponding value of n until n is an integer.

```
import math
t = int(input("Enter no. of cards"))
k = 1
n = math.log(1 + k*(t - 1), 2)
while n % 1 != 0:
    k += 1
    n = math.log(1 + k*(t - 1), 2)
print(f'This deck needs {int(n)} shuffles. k is {k}.')
```