An ‘in shuffle’ is a perfect shuffle on a standard deck of 52 playing cards that splits the deck in half, then interleaves cards starting with the top half.

\* What is the position of the first card after the 7th shuffle?

\* How many times must one perform the shuffle so that the top card becomes the bottom card?

\* When do the first and last cards in the deck touch?

I used Python to implement it. Here is the source code:

**Q1:** What is the position of the first card after the 7th shuffle?

**Ans:** If we start from the top half then first card remains at the same top position.

from random import shuffle

class Card:

    def \_\_init\_\_(self, value, suite):

        self.value = value

        self.suite = suite

def deck():

    suites = ['heart', 'diamonds', 'spades', 'clubs']

    deck = [Card(value, suite) for value in range(1, 14) for suite in suites]

    return deck

def in\_shuffle(deck):

    deck1,deck2=deck[0:int(len(deck)/2)],deck[int(len(deck)/2):len(deck)]

    new\_deck=[val for pair in zip(deck1, deck2) for val in pair]

    return new\_deck

if \_\_name\_\_ == "\_\_main\_\_":

    my\_deck=deck()

    shuffle(my\_deck)

    # for data in my\_deck:

    #     print(data.value,data.suite)

    ################### QUESTION#1 ###################################

    for i in range(0,7):

        shuffled\_deck=in\_shuffle(my\_deck)

        print(shuffled\_deck[0].value,shuffled\_deck[0].suite)

    print("1st card remains at the same position")

**Q2:** How many times must one perform the shuffle so that the top card becomes the bottom card?

**Ans:** If we will reverse the top half of deck upside-down then it can become the bottom card.

**Q3:** When do the first and last cards in the deck touch?

**Ans:** First and last cards won’t touch because if we start from the top half first card remains at the top position.

**In-shuffle Default:**

If we do not start from the top half.

If default in shuffle technique is used:

1. After the 7th shuffle the top card will be at 21st position.
2. It takes N/2 moves to bring first card at the bottom.
3. First and last card won’t touch.
4. from random import shuffle
5. class Card:
6. def \_\_init\_\_(self, value, suite):
7. self.value = value
8. self.suite = suite
9. def deck():
10. suites = ['heart', 'diamonds', 'spades', 'clubs']
11. deck = [Card(value, suite) for value in range(1, 14) for suite in suites]
12. return deck
13. def in\_shuffle(deck):
14. deck1,deck2=deck[0:int(len(deck)/2)],deck[int(len(deck)/2):len(deck)]
15. new\_deck=[val for pair in zip(deck2, deck1) for val in pair]
16. return new\_deck
17. if \_\_name\_\_ == "\_\_main\_\_":
19. my\_deck=deck()
20. shuffle(my\_deck)
21. # for data in my\_deck:
22. #     print(data.value,data.suite)
23. ################### QUESTION#2 ###################################
24. print(my\_deck[0].value,my\_deck[0].suite)
25. shuffled\_deck=in\_shuffle(my\_deck)

28. for i in range(0,25):
29. shuffled\_deck=in\_shuffle(shuffled\_deck)
31. print(shuffled\_deck[len(shuffled\_deck)-1].value,shuffled\_deck[len(shuffled\_deck)-1].suite)